

Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

« » 20 p.

Київ – 2021 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра автоматики та управління в технічних системах

Рівень вищої освіти – другий (магістерський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Програмне забезпечення інформаційно-комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«___» _____ 20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Бессмертному Роману Сергійовичу

1. Тема дисертації «Система управління наземною рухомою платформою на основі фреймворку Spring», науковий керівник дисертації Катін Павло Юрійович, к.т.н., доцент, затверджені наказом по університету від затверджені наказом по університету від «12» березня 2021 р. №809-с
2. Термін подання студентом дисертації 11.05.2021
3. Об'єкт дослідження: прототип автоматизованої рухомої платформи для дослідження небезпечних зон.
4. Предмет дослідження автоматизована система управління рухомою платформою на основі фреймворку Spring.
5. Перелік завдань, які потрібно розробити: аналіз варіантів реалізації системи управління рухомою платформою предметної області, створення апаратної інфраструктури, розроблення програмного рішення, графічний матеріал.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу: діаграма прецедентів, діаграма розгортання, діаграма компонентів, структурна діаграма, функціональна діаграма, діаграма сутностей, діаграма послідовності виконання команд, діаграма послідовності оброблення зображення, діаграма станів.

7. Орієнтовний перелік публікацій «Стандартизація, сертифікація, якість», «Проблеми інформатизації та управління», «Summer InfoCom Advanced Solutions 2021».

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 01.02.2021

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Огляд предметної області	01.02.2021	
2.	Аналіз існуючих рішень	08.02.2021	
3.	Аналіз та вибір технічних засобів для реалізації системи	15.02.2021	
4.	Розгортання середовища для розробки та тестування	22.02.2021	
5.	Розроблення програмного рішення	26.03.2021	
6.	Тестування розробленої системи	12.04.2021	
7.	Розроблення стартап – проекту	19.04.2021	
8.	Оформлення текстової документації	26.04.2021	
9.	Подання готової роботи	11.05.2021	

Студент

Роман БЕССМЕРТНИЙ

Науковий керівник

Павло КАТІН

РЕФЕРАТ

Магістерська дисертація на тему «Система управління наземною рухомою платформою на основі фреймворку Spring».

Робота містить 103 с. тексту, 16 рисунків, 20 таблиць, 22 джерела та 11 додатків.

Актуальність тематики зумовлена збільшенням потреби на рухомі автоматизовані системи. Результатом дослідження став прототип рухомої наземної платформи який дозволяє ефективно виконувати пошукові роботи, зменшуючи небезпеку для людини.

Об'єктом дослідження є прототип автоматизованої рухомої платформи для дослідження небезпечних зон.

Метою магістерської дисертації є зменшення трудомісткості (вартості) розробки програмної складової системи управління.

Поставлена мета досягається за рахунок використання загальнодоступного фреймворку, а також моделювання системи резервування з використанням математичного апарату марковських процесів.

Предметом дослідження є автоматизована система управління рухомою платформою на основі фреймворку Spring.

У ході роботи було створено прототип програмного застосунку з застосуванням мов програмування Java та C, та фреймворку Spring з конфігурацією запуску на платформі OrangePiZeroPlus.

Були проведено моделювання системи резервування з використанням математичного апарату марковських процесів. Було проведено дослідження відмовостійкості, розрахунки з використанням розробленої моделі.

Ключові слова: OrangePiZeroPlus, рухома платформа, відмовостійкість, кінцевий автомат.

ABSTRACT

Master's dissertation on "Ground mobile platform control system based on the Spring framework".

The work contains 103 pages of text, 16 drawings, 20 tables, 22 sources and 11 appendices.

The increasing needs of users in mobile automated systems create a relevancy for the research on the topic. The result of the study was a prototype of mobile ground platform systems that can be used to search for exploration, reducing the danger to humans.

The object of research is a prototype of an automated mobile platform for the study of dangerous areas.

The purpose of the master's dissertation is to reduce the complexity (cost) of developing a software component of the management system.

This goal is achieved through the use of a publicly available framework, as well as the modeling of redundancy systems using the mathematical apparatus of Markov processes.

The subject of the study is an automated mobile platform control system based on the Spring framework.

During the work, a prototype of a software application using Java and C programming languages and Spring framework, deployed on the OrangePiZeroPlus platform, was created.

Modeling of redundancy systems was performed using the mathematical apparatus of Markov processes. A study of fault tolerance and the calculations using the developed models were conducted.

Keywords: OrangePiZeroPlus, mobile platform, fault tolerance, state machine.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ.....	9
ВСТУП.....	10
1 АНАЛІЗ ВАРІАНТІВ РЕАЛІЗАЦІЇ СИСТЕМИ УПРАВЛІННЯ РУХОМОЮ ПЛАТФОРМОЮ.....	12
1.1 Загальна характеристика автоматизованих систем управління.....	12
1.1.1 Системи управління та їх структура.....	12
1.1.2 Автоматичне і автоматизоване управління.	15
1.1.3 Автоматизовані системи управління	15
1.1.4 Життєвий цикл АС	16
2 АПАРАТНА ІНФРАСТРУКТУРА СИСТЕМИ УПРАВЛІННЯ НАЗЕМНОЮ РУХОМОЮ ПЛАТФОРМОЮ.....	19
2.1 Одноплатні комп'ютери як основа системи управління рухомою платформою.	20
2.2 Вибір одноплатного мікрокомп'ютера для реалізації основного керівного модуля системи управління рухомою платформою.	21
2.2.1 Raspberry Pi.....	22
2.2.2 Rock Pi 4 Модель В.....	23
2.2.3 BeagleBone Black	25
2.2.4 Nvidia Jetson Nano.....	25
2.2.5 ASUS Tinker Board S	27
2.2.6 Orange Pi Zero Plus.....	28
2.2.7 Обґрунтування вибору Orange Pi Zero Plus для реалізації основного керівного модуля системи управління рухомою платформою. ..	29

2.3 Порівняння характеристик інтерфейсів бездротового зв'язку для системи управління рухомою платформою	30
2.3.1 Критерії оцінки бездротових комунікаційних протоколів.....	31
2.3.2 Wi-Fi.....	32
2.3.3 802.15 - Бездротові персональні мережі.....	32
2.3.4 802.15.1 - WPAN / Bluetooth	32
2.3.5 802.15.4	32
2.3.6 Протоколи передачі стільникових даних	32
2.3.7 LoRa.....	33
2.3.8 Sigfox.....	33
2.3.9 Z-Wave	33
2.3.10 RFID та NFC	33
2.3.11 Діаграма результативності.....	33
2.3.12 Перспективні комунікаційні протоколи.....	35
2.3.13 Висновки.....	35
2.4 Вибір відеокамери для системи управління рухомою платформою	36
2.5 Огляд шасі для мобільних платформ	37
2.6 Вибір апаратних компонентів системи управління виконавчими механізмами рухомої платформи.	39
2.6.1 STM32F401 як мікроконтролер для управління виконавчими механізмами в режимі реального часу.....	40
2.6.2 Використання контролеру L298N H-bridge для управління двигунами системи управління рухомою платформою	41
2.7 Джерело живлення	42

2.7.1 Види джерел живлення, що використовуються в робототехніці	42
2.7.2 Порівняння Li-Ion, Li-Poly і NiMH	43
2.7.3 Аналіз та обґрунтування обраної системи живлення.	44
2.8 Розробка структурної схеми системи управління рухомою платформою	47
2.9 Розробка функціональної схеми системи управління рухомою платформою	49
2.10 Висновки	52
3 ФЕРЙМВОРК SPRING ЯК ОСНОВА ПРОГРАМНОЇ ЧАСТИНИ СИСТЕМИ УПРАВЛІННЯ РУХОМОЮ ПЛАТФОРМОЮ	53
3.1 Ферймворк Spring як основа системи управління рухомою платформою	53
3.2 Переваги фреймворка Spring як програмної основи системи управління рухомою платформою	53
3.3 Недоліки фреймворку Spring як фактори що обмежують програмну функціональність ферймворку у складі системи управління.....	55
3.4 Система передачі відеоінформації користувачу рухомої платформи на основі Video4Linux4Java	56
3.5 Інтегрована база даних для збереження інформації рухомої платформи	57
3.6 Діаграма компонентів системи управління рухомої платформи на основі фреймворку Spring	59
3.7 Висновки	61
4 ПРОГРАМНА ІНФРАСТРУКТУРА СИСТЕМИ УПРАВЛІННЯ НАЗЕМНОЮ РУХОМОЮ ПЛАТФОРМОЮ.....	62

4.1 Сценарії використання системи	63
4.2 Система передачі команд управління рухомою платформою	63
4.3 Система передачі поточного відео	77
4.4 Інтегрована система збереження даних системи управління рухомої платформи	78
4.5 Моделювання відмовостійкості за допомогою марковських процесів	78
5 СТАРТАП ПРОЕКТ	86
5.1 Опис ідеї проекту	86
5.2 Технологічний аудит ідеї проекту.....	89
5.3 Аналіз ринкових можливостей запуску.....	90
5.4 Ринкова стратегія	98
5.5 Висновки	99
ВИСНОВКИ	100
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	101

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

СУРП – Система управління рухомою платформою

СУ – Система управління

САУ – система автоматичного управління

АСУ – автоматизована система управління

ОПК – одноплатний комп'ютер

АЦП – аналого-цифровий перетворювач

ШІМ – широтно-імпульсна модуляція

GPIO – General Purpose Input-Output

EXTI – External interrupt/event controller

ВСТУП

При виконанні пошукових робіт на небезпечних для людини ділянках місцевості(приміщеннях) використовуються дистанційно керовані мобільні роботи. В даний час в світі це найбільш поширене застосування наземних колісних і гусеничних мобільних платформ[1]. Основна перевага, яке дає їх застосування - можливість розмістити фахівців, в даному випадку оператора мобільної платформи, поза межами потенційно небезпечної зони, зберегти здоров'я і життя людей. Проблема виникає при втраті зв'язку, погіршенні видимості і інших випадках, коли не може використовуватися ручне управління. Наприклад, пожежна мобільна платформа виконує дослідження території хімічного заводу після пожежі. Якщо, перебуваючи далеко від оператора, мобільна платформа заїде за масивні металеві конструкції, може статися повна або часткова втрата зв'язку. Те ж відбувається при радіаційному опроміненні або пошкодженні приймально-передавача. Крім того, при задимленні оператору складно орієнтуватися по зображень бортових телекамер. Все це ускладнює повернення мобільної платформи з несприятливих умов, і як результат може призвести до пошкодження або втрати дорогого обладнання. При відсутності дистанційного управління застосовуються системи автоматичного повернення робота до оператора, які засновані на використанні автономної навігації.

Актуальність роботи полягає у збільшенні потреби до автоматизації дослідження небезпечних зон за допомогою рухомих платформ та високі вимоги до відмовостійкості у цій предметній області.

Об'єктом дослідження є прототип автоматизованої рухомої платформи для дослідження небезпечних зон.

Предметом дослідження є автоматизована система управління рухомою платформою на основі фреймворку Spring.

Метою магістерської дисертації є зменшення трудомісткості (вартості) розробки програмної складової системи управління.

Поставлена мета досягається за рахунок використання загальнодоступного фреймворку, а також моделювання системи резервування з використанням математичного апарату марковських процесів.

Конкретні завдання роботи:

- аналіз варіантів реалізації системи;
- розробка апаратної інфраструктури автоматизованої системи управління мобільною платформою;
- розробка програмної інфраструктури автоматизованої системи управління мобільною платформою;
- проведення експериментального дослідження.

Наукова новизна полягає в тому, що було проведено дослідження впливу резервування на відмовостійкість автоматизованої системи управління рухомою платформи, що побудована на базі фреймворка Spring. Результат дослідження показав що резервування дозволяє зменшити вартість експлуатації систему управління.

Апробація роботи. Результати виконаної дослідницької роботи були опубліковані у міжнародній науково-практичній конференції «Summer InfoCom Advanced Solutions 2021» у доповіді «Використання високопродуктивних мікроконтролерів для автоматизованої платформи дослідження території» у 2021 році в м. Києві.

Публікації. Результати виконаної дослідницької роботи були опубліковані у статті «Використання високопродуктивних мікроконтролерів для підвищення економічної ефективності виробництва джему» в науковому журналі «Стандартизація, сертифікація, якість» у 2019 році в м. Києві.

1 АНАЛІЗ ВАРІАНТІВ РЕАЛІЗАЦІЇ СИСТЕМИ УПРАВЛІННЯ РУХОМОЮ ПЛАТФОРМОЮ

У даному розділі розглядаються принципи побудови автоматизованих систем управління в контексті використання як основи СУРП.

1.1 Загальна характеристика автоматизованих систем управління

В даному підрозділі розглядаються системи управління, їх структура, основні властивості автоматизованих систем управління та їх життєвий цикл в контексті використання для СУРП.

1.1.1 Системи управління та їх структура

Задачі автоматизації зводяться до створення систем управління машинами, агрегатами, поточними лініями, наземними об'єктами, повітряними об'єктами тощо[2, 4–6].

Відповідно до загального визначання СУ у процесі управління існує об'єкт, яким управляють і система, що здійснює управління (технічний засіб, людина). У процесі управління цей модуль отримує інформацію про стан зовнішнього середовища, де перебуває об'єкт і з яким він пов'язаний. Уся ця інформація сприймається управляючим модулем, який виробляє на її основі управляючу інформацію (приймає рішення). На основі прийнятого рішення виконавчий модуль здійснює управляючий вплив на об'єкт управління.

Структурна схема системи управління наведена на рис.1.1.

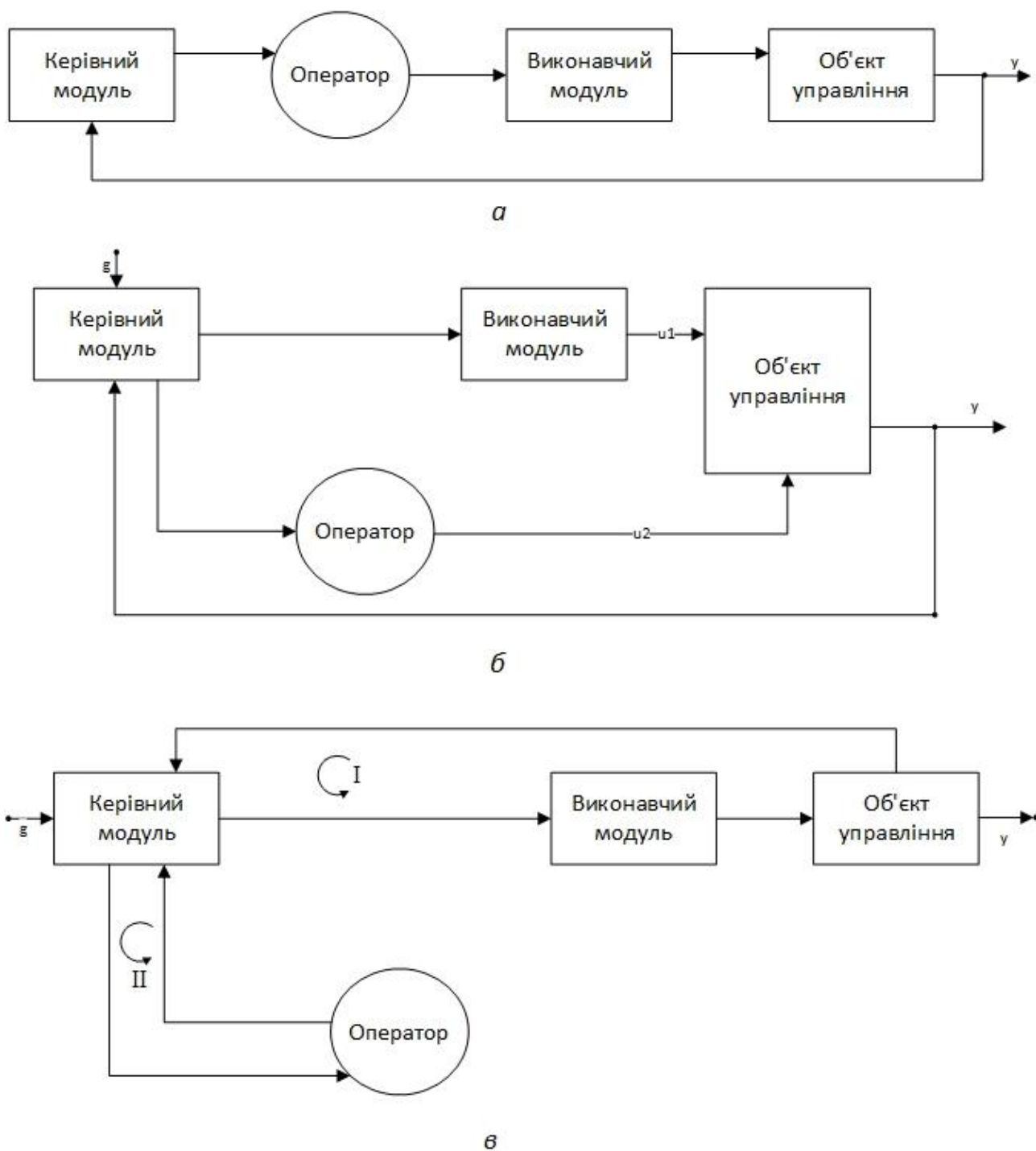


Рис. 1.1 Структура системи управління[2]

Об'єкт управління (ОУ) – умовно виокремлена частина системи, на яку впливає система управління для досягнення необхідного результату.

Управління завжди здійснюється для досягнення зазначеної мети, яка завжди конкретна для заданого об'єкта управління і пов'язана зі станом об'єкта та середовища, в якому він перебуває.

Серед усіх видів інформації, що надходить до управляючого модулю, надзвичайно важлива інформація, яка йде по лініях зворотного зв'язку від об'єкта управління. У системах управління зворотний зв'язок визначається як інформаційний зв'язок, за допомогою якого в управляючу частину надходить інформація про наслідки управління об'єктом, тобто інформація про новий стан об'єкта, який виник під впливом управляючих дій.

Відповідно до структури системи управління, зображеної на рисунку 1.1, людина може виконувати різні функції. На рисунку 1.1, *а* показана система де керівний модуль отримує інформацію про стан ОУ, оброблює її і передає людині-оператору у вигляді, зручному для прийняття рішення. Людина у цій системі виконує функції прийняття рішень(фактично керівного модуля). Один із варіантів такого способу розподілення функцій представлений на рисунку 1.1, *б*.

У цьому випадку керівний модуль отримує і оброблює інформацію про всі параметри ОК і виконує керування деякими з них(u_1). Людина-оператор отримує від керівного модуля інформацію, яку вона використовує для управління іншими параметрами(u_2). У двох розглянутих варіантах контур управління був розірваний, і замикався за допомогою людини-оператора. У схемі, зображеній на рис 1.1, *в*, основний контур керування є замкнутим. Керівний модуль управляє параметрами ОК згідно з закладеному алгоритму. При зміні умов роботи ОК людина-оператор змінює програму роботи керівного модуля. У даному випадку оператор та керівний модуль утворюють другий контур системи(позначений цифрою II). Це є прикладом адаптивної системи, у якій контур II використовується в міру необхідності. Такий контур виконує функцію пам'яті – досвід людини-оператора.

Більшість існуючих автоматизованих СУРП мають структури, аналогічні зображеним на рисунку 1.1, *а*, *б*, як, наприклад, у роботі[10]. У даній роботі буде

розглянута можливість та доцільність використання автоматизованої системи управління з замкнутим основним контуром керування(рисунок 1.1, в).

1.1.2 Автоматичне і автоматизоване управління.

Системи управління поділяються на два класи [2, 4–6]: системи автоматичного управління (САУ) і автоматизовані системи управління (АСУ). У САУ управління об'єктом або системою здійснюється автоматичними пристроями без безпосередньої участі людини.

Основні функції САУ: автоматичний контроль і вимірювання, автоматична сигналізація, автоматичний захист, автоматичний пуск і зупинка різних двигунів і приводів, автоматична підтримка заданих режимів роботи устаткування, автоматичне регулювання. На відміну від САУ в АСУ у сферу управління включена людина, на яку покладаються функції прийняття найважливіших рішень і відповідальності за прийняті рішення.

АСУ є людино-машинними системами, що використовують сучасні економіко-математичні методи, засоби електронно-обчислювальної техніки, а також нові організаційні принципи для пошуку і реалізації ефективного управління об'єктом. У роботі АСУ необхідна участь людини - оператора.

1.1.3 Автоматизовані системи управління

Автоматизована система управління – автоматизована система (АС), призначена для автоматизації процесів збирання та пересилання інформації про об'єкт управління, її перероблення та видачі керівних дій на об'єкт управління [6].

Найважливіше завдання АСУ – підвищення ефективності управління об'єктом на основі зростання продуктивності праці та вдосконалення методів планування процесу управління.

Функції АСУ встановлюють у технічному завданні на створення конкретної АСУ на основі аналізу цілей управління, заданих ресурсів для їх досягнення, очікуваного ефекту від автоматизації і відповідно до стандартів, що поширюються на цей вид АСУ [7].

Функції АСУ в загальному випадку включають в себе наступні елементи:

- планування і прогнозування;
- облік, контроль, аналіз;
- координацію і регулювання.

АСУ повинна здійснювати наступні дії:

- збирання, обробку і аналіз інформації (сигналів, повідомлень, документів і т. п.) про стан об'єкту управління;
- вироблення управляючих дій (програм, планів і т. п.);
- передавання управляючих дій (сигналів, вказівок, документів) на виконання;
- реалізація і контроль виконання управляючих дій;
- обмін інформацією (документами, повідомленнями і т. п.) зі взаємозв'язаними автоматизованими системами.

Приклади АСУ: автоматизована система управління дорожнім рухом, автоматизована система управління вуличним освітленням, автоматизована система управління підприємством, система управління готелем.

1.1.4 Життєвий цикл АС

Життєвий цикл автоматизованої системи – сукупність взаємопов'язаних процесів створення та зміни стану АС від формування початкових вимог до неї до закінчення експлуатації та утилізації [6].

Життєвий цикл передбачає наступні стадії і етапи:

1. Формування концепції АС:

- обстеження об'єкту і обґрунтування необхідності створення АС; формування вимог користувача до АС; оформлення звіту про виконання робіт та заявки на розробку АС; вивчення об'єкта; проведення необхідних науково-дослідних робіт;

- аналіз матеріалів і формування документації: розробка варіантів концепції АС і вибір варіанта концепції АС, що задовольняє вимогам користувачів; оформлення звіту про виконану роботу, розробка та затвердження технічного завдання на створення АС.

2. Розробка:

- ескізне проектування – розробка попередніх проектних рішень щодо систем і її частин; розробка документації на АС і її частини;

- технічне проектування: розробка проектних рішень щодо системи і її частин; розробка документації на АС і її частини; розробка та оформлення документації на поставку комплектуючих виробів; розробка завдань на проектування в суміжних частинах проекту.

3. Реалізація:

- розробка робочої документації на АС і її частини; створення робочого проекту; підготовка інструкцій користувачів;

- програмування: розробка програмних модулів; тестування; коректування бази даних.

4. Експлуатація:

- підготовка до введення в дію: підготовка об'єкта автоматизації, підготовка персоналу, комплектація АС програмними і технічними засобами, програмно-технічними комплексами, інформаційними виробами, будівельно-монтажні роботи, пусконаладжувальні роботи, проведення попередніх випробувань;

- проведення дослідної експлуатації;

– проведення приймальних випробувань.

5. Супровід АС: супровід програмних засобів. Оперативне обслуговування, адміністрування базами даних.

6. Зняття з експлуатації: припинення використання, демонтаж, архівування системи.

1.2 Висновок

У цьому розділі розглядаються теоретичні основи розробки автоматизованих систем управління, у тому числі СУРП.

Результати огляду загальних характеристик автоматизованих систем управління, а також існуючих різновидів систем управління будуть використані у ході розробки структурних елементів програмної та апаратної інфраструктури СУРП.

У першій частині розділу була представлена загальна структура систем управління, були порівняні автоматичні і автоматизовані системи управління. Були розглянуті основні характеристики та завдання автоматичних систем управління.

Крім того, проведено огляд життєвого циклу автоматизованої системи, і розглянуто усі стадії, які необхідно виконати для впровадження автоматизованої СУРП.

2 АПАРАТНА ІНФРАСТРУКТУРА СИСТЕМИ УПРАВЛІННЯ НАЗЕМНОЮ РУХОМОЮ ПЛАТФОРМОЮ

У минулій главі були теоретично розглянуті методи керування наземною рухомою платформою, зокрема метод візуальної одометрії, що дозволяє вирахувати положення рухомої платформи за допомогою системи тривимірного технічного зору. Для тестування та розробки такої системи в реальних умовах потрібно побудувати її прототип, після чого можна експериментально оцінити характеристики та надійність такої системи. Цей розділ розгляне усі компоненти прототипу з точки зору їх взаємодії з навколишнім світом та іншими компонентами системи.

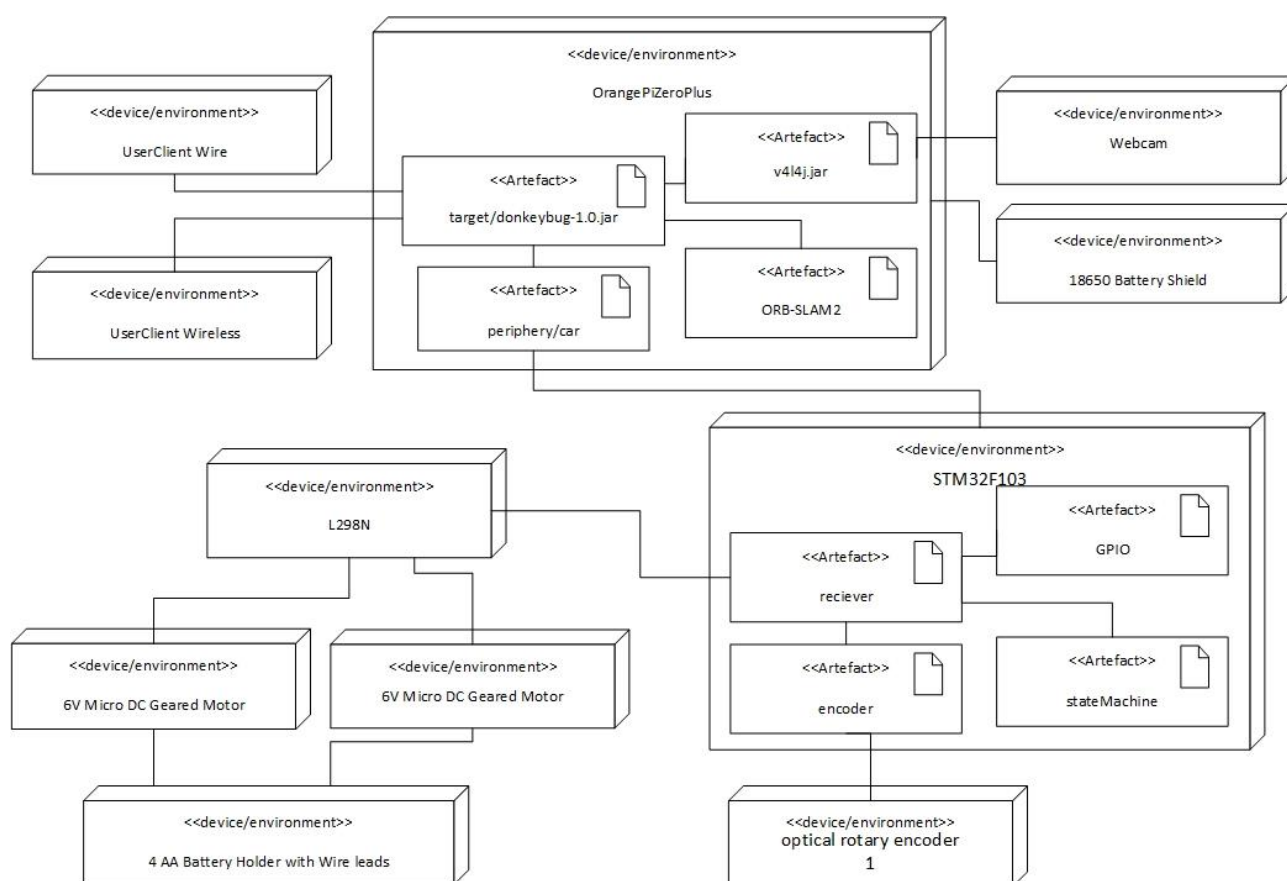


Рисунок 2.1 – Діаграма розгортання апаратної інфраструктури

На рис 2.1 показана діаграма розгортання наземного робота. Принципово систему можна поділити на два рівні – стратегічний і тактичний.

2.1 Одноплатні комп'ютери як основа системи управління рухомою платформою.

Для основи СУРП доцільно використати одноплатні комп'ютери(ОПК). Це обумовлено наступними причинами:

- висока потужність одноплатних комп'ютерів що на теперішній час наближується до потужності персональних комп'ютерів;
- прийнятна вартість(до \$200) на час написання роботи;
- можливість використання операційної системи на борту ОПК для реалізації програмної частини СУРП.

Спеціалізовані рухомі платформи, можуть виготовляються в партіях з великими і невеликими кількостями. Тому економічно доцільним є використання ОПК для реалізації апаратної частини СУРП. Використання ОПК дозволяє користуватися одним і тим же комп'ютером у багатьох шасі або конструкціях - можна просто оновити мозок системи, коли з'являється більш потужне обладнання.

Якщо робот повинен рухатися самостійно, щоб виконувати свої функції, ймовірно доведеться використовувати можливість машинного зору. На жаль, це вимагає сильних обчислювальних можливостей, і знайти підтримку для увімкнення машинного зору може бути важко. Однак ця функція має вирішальне значення, щоб ефективно розуміти динамічне середовище та керувати ним. Статичне середовище створює мало навігаційних проблем і може мати фіксовані шляхи, але коли середовище робота вимагає від нього роботи з людьми, воно повинно мати можливість реагувати на несподіване.

Також потрібно враховувати різні методи управління рухом. ШІМ-контролер може взаємодіяти з ОПК, щоб подавати двигунам імпульси потужності, контролюючи рух і швидкість робота. Існуючі способи застосування ШІМ-контролю будуть детально розглянуті далі. Широтна-імпульсна модуляція - це чудовий спосіб регулювання управління двигуном, і, як правило, забезпечує

найкращу ефективність, але швидкість також можна регулювати для невеликих двигунів за допомогою простих апаратних засобів, таких як потенціометр.

Датчики та інтерфейси датчиків надзвичайно корисні для роботів, яким потрібно визначити що-небудь щодо свого оточення, наприклад, їх орієнтацію чи напрямок. Акселерометри, гіроскопи та магнітометри надають роботам різну інформацію про їх фізичну ситуацію і, отже, про здатність рухатися та розташуватися в реальності.

Деякі дошки вже мають пристойний набір датчиків, але якщо у ОПК є доступні шини SPI або I2C, існує можливість легко додати ще більше. Вбудовані аналого-цифрові перетворювачі (АЦП) дають можливість використовувати аналогові датчики, які, як правило, мають швидший відгук і є менш дорогими, ніж цифрові датчики.

2.2 Вибір одноплатного мікрокомп'ютера для реалізації основного керівного модуля системи управління рухомою платформою.

Існує безліч міркувань при виборі ОПК, який найкраще відповідає вимогам СУРП. Популярність ОПК вплине на доступні ресурси спільноти, такі як форуми, книги, навчальні посібники та апаратні аксесуари. Необхідно також проаналізувати інтерфейси підключення. Наприклад, піни GPIO, Bluetooth, Wi-Fi та роз'єми для камер, двигунів, РК-екранів та інших компонентів, необхідних для роботи системи. І головним фактором є ціна. Хоча багато ОПК коштують близько 35 доларів США і залишаються десь нижче 100 доларів, інші продаються за кілька сотень доларів.

Міркування щодо ОПК для СУРП:

- ресурси та розміри громади;
- ціна;
- доступні готові набори для робототехніки;
- входи / виходи (I / O), тобто піни GPIO тощо;

– сумісність програмного та апаратного забезпечення;

2.2.1 Raspberry Pi

Raspberry Pi, можливо, є одним з найкращих ОПК на ринку. Існує безліч навчальних посібників, які пояснюють, як використовувати Raspberry Pi як баггі, робота, готового до змагань, або інших подібних проектів. Крім того, існує безліч готових наборів роботів на основі Raspberry Pi, такі як SunFounder PiCar-S або PiBorg. Цей ОПЕ чудово задокументований з великою базою користувачів, тому існує велика кількість підручників, які можна використати для навчання та форуми, де можна звернутися за допомогою щодо усунення несправностей. Raspberry Pi Zero / Zero W може працювати як невеликий ОПК опція з низьким енергоспоживанням. Або існує Raspberry Pi 3 B + або Pi 4, якщо необхідна додаткова обчислювальна потужність.

Переваги Raspberry Pi 4:

- Доступний
- Безліч моделей на вибір
- Доступні готові набори робототехніки
- Відмінна сумісність апаратного та програмного забезпечення
- Велика громада

Технічні характеристики Raspberry Pi 4:

- Broadcom BCM2711, чотириядерний Cortex-A72 (ARM v8) 64-розрядний SoC @ 1,5 ГГц
- 1 ГБ, 2 ГБ, 4 ГБ або 8 ГБ LPDDR4-2400 SDRAM (залежно від моделі)
- 2,4 ГГц та 5,0 ГГц IEEE 802.11ac бездротовий, Bluetooth 5.0, BLE
- Гігабітний Ethernet
- 2 порти USB 3.0; 2 порти USB 2.0.
- Стандартний 40-контактний роз'єм GPIO Raspberry Pi (повністю зворотно сумісний з попередніми платами)

- 2 порти micro-HDMI (підтримується до 4кp60)
- 2-смуговий порт дисплея MIPI DSI
- 2-смуговий порт камери MIPI CSI
- 4-полюсний стереозвук та композитний відеопорт
- H.265 (декодування 4кp60), H264 (декодування 1080p60, кодування 1080p30)
- Графіка OpenGL ES 3.0
- Слот для карт MicroSD для завантаження операційної системи та зберігання даних
- 5 В постійного струму через роз'єм USB-C (мінімум 3 А *)
- 5 В постійного струму через роз'єм GPIO (мінімум 3 А *)
- Можливе живлення через Ethernet (PoE) (потрібен окремий PoE HAT)
- Робоча температура: 0-50 градусів Цельсія навколишнього середовища

2.2.2 Rock Pi 4 Модель B

Модель Rock Pi 4 Model B, що працює на потужній системі на чіпі Rockchip RK3399 (SoC), є переконливою альтернативою Raspberry Pi 4. Шість-ядерний процесор ARM, 64-розрядна двоканальна оперативна пам'ять LPDDR4 3200 Мбіт / с та вихід HDMI з роздільною здатністю 4K зі швидкістю 60 кадрів в секунду (FPS) є стандартними. Цей ОПК оснащений чудовими вбудованими пінами вводу / виводу, інтерфейсом дисплея MIPI (DSI), послідовним інтерфейсом камери MIPI (CSI), 3,5-мм роз'ємом для навушників, Wi-Fi 802.11 ac, Bluetooth 5.0, гігабітною локальною мережею, та годинником реального часу (RTC).

Завдяки своєму 40-контактному інтерфейсу GPIO, Rock Pi 4 надзвичайно зручний для виробників. Ви можете підключити кнопки, датчики, світлодіоди, перемикачі, двигуни тощо. Більше того, його Bluetooth та Wi-Fi надають широкі

можливості підключення. Можна навіть підключити камери, використовуючи роз'єм MIPI CSI.

Переваги Rock Pi 4 Model:

- Відмінне введення / виведення
- Доступна ціна
- Потужний SoC

Характеристики Rock Pi 4:

- Процесор: Rockchip RK3399 подвійний Cortex-A72, @ 1,8 ГГц, чотириядерний Cortex-A53 @ 1,4 ГГц
- Графічний процесор: графічний процесор Mali T860MP4 з підтримкою OpenGL ES 1.1 /2.0 /3.0 /3.1 /3.2, Vulkan 1.0, Open CL 1.1
- 1.2, DX11
- 1 ГБ, 2 ГБ або 4 ГБ 64-розрядної двоканальної оперативної пам'яті

LPDDR4

- 40-контактний роз'єм GPIUO
- Модуль eMMC
- Роз'єм SSD M.2
- HDMI 2.0 до 4K @ 60
- MIPI DSI
- Гніздо 2,5 мм
- MIPI CSI
- Wi-Fi 802.11 ac, Bluetooth 5.0
- USB 3.0 з OTG, USB 2.0
- Гігабітний LAN

2.2.3 BeagleBone Black

Незважаючи на посередні характеристики - 512 МБ оперативної пам'яті DDR3, 4-бітний 8-бітний модуль eMMC, прискорення тривимірної графіки, подвійні 32-розрядні мікроконтролери PRU та прискорювач із плаваючою точкою NEON - BeagleBone Black, тим не менш, служить міцною основою для різних робототехнічних проектів. З боку програмного забезпечення існує можливість запускати операційні системи Linux, такі як Ubuntu та Debian, а також Android та безліч інших ОС.

Також є першокласне підключення, включаючи HDMI, Ethernet, USB-хост, USB-клієнт для зв'язку та живлення, а також два 46-контактних заголовки. Унікальною особливістю BeagleBone Black є його велика спільнота, величезні ресурси та апаратні доповнення. Існує велика кількість форумів, спільнот та книг, які спрощують роботу. Більше того, BeagleBone пропонує безліч різних апаратних додатків.

Переваги BeagleBone Black:

- Доступний
- Відмінна сумісність апаратного та програмного забезпечення
- Доступні накідки BeagleBoard, включаючи накідку робототехніки
- Подвійні 46-контактні роз'єми GPIO

2.2.4 Nvidia Jetson Nano

Nvidia Jetson – є одним із найпотужніших ОПК для робототехніки, спеціально розроблений для роботи з (ШІ). Коштує трохи більше 300 доларів, це не є дешевим варіантом, але цей ОПК надає і багато можливостей. Тоді як для більшості ОПК існують роботизовані набори, NVIDIA пропонує вже готову рухому платформу з вбудованими системами автоматичної обробки візуальної інформації. Збудований для досліджень автономних транспортних засобів, ця

система наразі є найбільш компетентним обладнанням для досліджень у галузі робототехніки.

Jetson Nano є open-source у всіх аспектах, як апаратному, так і програмному. Це значно спрощує використання Jetson як відправної точки для створення нових роботів на основі штучного інтелекту. На сторінці GitHub викладено все, що потрібно щоб розпочати роботу, а також навчальні посібники з основ, зокрема система запобігання зіткненням на основі штучного інтелекту. Крім того, Nvidia Jetson розвиває підтримку розповсюдження Linux для настільних комп'ютерів та забезпечує 472 GFLOPS обчислювальної продуктивності при збереженні низької потужності у 5 Вт. У той же час не має необхідності використовувати саме JetBot, оскільки використовуючи даний ОПК можна зібрати проект з нуля.

Переваги Nvidia Jetson Nano:

- Найпотужніший ОПК на ринку
- Доступна готова автономна конструкція автомобіля на основі AI з відкритим вихідним кодом
- Спрощене впровадження систем з штучним інтелектом
- Велика підтримка від основних та сторонніх джерел

Характеристики Nvidia Jetson Nano:

- Графічний процесор: 128-ядерний графічний процесор на базі архітектури NVIDIA Maxwell™
- Процесор: чотириядерний ARM® A57
- Відео: 4K @ 30 fps (H.264 / H.265) / 4K @ 60 fps (H.264 / H.265) кодування та декодування
- Камера: смуги MIPI CSI-2 DPHY, 12x (модуль) та 1x (комплект розробника)
- Пам'ять: 4 Гб 64-розрядна LPDDR4; 25,6 гігабайт / сек
- Підключення: гігабітний Ethernet
- Підтримка ОС: Linux для Tegra®
- Розмір модуля: 70 мм x 45 мм

- Розмір комплекту розробника: 100 мм x 80 мм

2.2.5 ASUS Tinker Board S

ASUS Tinker Board і Tinker Board S пропонують велику кількість функцій за середньою ціною. Вироблені в тому ж форм-факторі, що і Raspberry Pi 3 та Pi 3 B +, Tinker Board має перевагу завдяки чудовій сумісності з корпусами та аксесуарами Raspberry Pi. Цей ОПК містить 40-контактний роз'єм GPIO та безліч варіантів підключення, включаючи I2C, S / PDIF, MIPI CSI, MIPI DSI, Wi-Fi, Bluetooth та GbE LAN.

Книга Practical Tinker Board надає збірку робіт у комплекті зі списком витратних матеріалів, який включає Adafruit Motor HAT. Для робототехніки можна розглянути Tinker Board S, оскільки, на відміну від звичайного Tinker Board, S має вбудований апаратний ШІМ для управління двигуном. Завдяки своїм розмірам та роз'єму GPIO Tinker Board та Tinker Board S є сумісними з більшістю аксесуарів Raspberry Pi, що робить цей ОПК надійною альтернативою.

Переваги ASUS Tinker Board S:

- Сумісний з аксесуарами Raspberry Pi 3/3 B +
- Відмінна сумісність апаратного та програмного забезпечення
- Велика підтримка громади

Характеристики ASUS Tinker Board S:

- Rockchip RK3288 Cortex-A17 Quad-Core SoC
- Двоканальний LPDDR3 на 2 Гб
- HDMI (підтримує до 4K)
- ARM Mali-T760 MP4 - підтримує OpenGL ES1.1 / 2.0 / 3.0, OpenVG1.1, OpenCL, DirectX11
- Слот для карт пам'яті eMMC та Micro SD (TF) на 16 Гб
- Realtek RTL8211E-VB-CG Гігабітний Ethernet
- 802.11 b / g / n, Bluetooth V4.0 + EDR

- RTL ALC4040 CODEC, що підтримує звук до 24 біт / 192 кГц
- Апаратне забезпечення, готове для виявлення роз'єму аудіороз'єму
- 4 x USB 2.0
- 1 x 40-контактний роз'єм, 1 x 2-контактний контактний контакт, 1 x 15-контактний MIPI DSI, 1 x 15-контактний MIPI CSI
- 3,37 "x 2,125" (8,55 см x 5,4 см)

2.2.6 Orange Pi Zero Plus

Orange Pi Zero Plus є ОПК, який містить мінімально необхідну кількість компонентів, які забезпечують його функціонування.

ОПК містить на собі роз'єми введення та виведення для створення апаратної інфраструктури. В даний час компанія продає голу ОПК — корпус не входить у комплект. Що ж стосується програмного забезпечення, в даний час є три операційних системи на основі Linux, підтримуваних Orange Pi Zero Plus. Що стосується специфікацій, то Orange Pi Zero Plus є пристроєм, розміром 48 x 46 мм. Система включає в себе 64-бітний процесор Allwinner H5 Quad-core Cortex-A53, що працює на частоті 1.2 ГГц, а також графічний процесор Неха-core Mali450. Вона має 512 МБ оперативної пам'яті DDR3. Orange Pi отримує енергію від зарядного пристрою Micro USB 5V 2A AC, і вимагає стабільної напруги для надійної роботи. У той час як ARM процесор забезпечує реальну продуктивність, Неха-core Mali450 GPU є потужним графічним ядром, здатним до апаратного декодування декількох форматів відео високої якості.

Переваги OrangePi Zero Plus:

- вдала комбінація невеликого розміру комп'ютера і доступної ціни
- апаратні засоби з повністю відкритим вихідним кодом
- наявність безкоштовної ОС “Armbian”
- сумісність даної ОС з технологіями JAVA[8]

— наявність безкоштовних, надійних бібліотек для виконання поставленої задачі

Технічні характеристики ОПК OrangePi Zero Plus [8]:

- процесор 64-бітний SoC Allwinner H5 Quad-core Cortex-A53;
- графічний процесор Mali450MP4;
- оперативна пам'ять 512Mb DDR3 synchronous dynamic random-access memory (включаючи GPU);
- вбудована пам'ять 8GB embedded MultiMediaCard;
- можливість підтримки карт пам'яті microSD (об'ємом до 64GB);
- аудіо— і відеовивід через HDMI з підтримкою Consumer Electronics Control;
- вбудований WiFi 802.11 b/g/n + Bluetooth 4.0 Low Energy;
- microUSB 2.0 On-The-Go x1;
- Camera Serial Interface;
- налагоджувальний порт;
- 26 пінів General Purpose Input-Output, сумісних з Raspberry Pi B +;
- світлодіоди стану і активності;
- живлення від microUSB роз'єму;
- розміри 48 x 46 mm;
- вага 20g.

2.2.7 Обґрунтування вибору Orange Pi Zero Plus для реалізації основного керівного модуля системи управління рухомою платформою.

Основний керівний модуль СУРП відповідає за обробку відеоінформації, створення карти місцевості, комунікацію з користувачем і забезпечення збереження інформації.

Оскільки ці задачі потребують інтенсивних обчислень, зокрема потужного блоку графічної обробки, а робота у реальному часі не є критичною, обчислення

стратегічного рівня виконуються за допомогою мікрокомп'ютера OrangePiZeroPlus.

Дослідження переваг використання багатоядерних високопотужних мікроконтролерів, зокрема мікрокомп'ютера OrangePiZeroPlus для виконання сучасних завдань автоматизації та управління в технічних системах було виконано у роботі[9-10]. Зокрема, переваги використання цієї системи для наземної рухомої платформи були досліджені у роботі[11].

З цих наукових досліджень можна зробити висновок, що використання ОПК OrangePiZeroPlus у програмно-апаратній інфраструктурі СУРП на основі фреймворку Spring має значні переваги, а саме низьку ціну, широкий набір інтерфейсів вводу-виводу інформації, потужні обчислювальні можливості, а також наявність доступної документації та стандартизованого підходу до створення автоматизованих систем управління, зокрема з елементами візуального контролю.

ОПК на основі архітектури Cortex-A мають специфічні вимоги до живлення, оскільки потребують доволі сильний струм у 3А і високу стабільність напруги. На основі наявної інформації[12], а також наукового дослідження[11], можна зробити висновок що для живлення мікрокомп'ютера OrangePiZeroPlus доцільно використовувати літій-іонний акумулятор 18650, та окрему плату 18650 Lithium Battery Charge Shield V3.

2.3 Порівняння характеристик інтерфейсів бездротового зв'язку для системи управління рухомою платформою

Історично СУРП керувалися за допомогою дротових інтерфейсів. Дротові інтерфейси мають істотні недоліки, особливо при використанні у небезпечних умовах. Вірогідність розриву дроту, і, відповідно, втрати зв'язку, є значною. Відповідно, було прийнято рішення використовувати бездротовий зв'язок. Згідно з предметною областю СУРП повинен мати можливість передавати відеодані в

реальному часі. Задача дослідження території обмежує мінімальну відстань зв'язку розмірами цієї території.

Незважаючи на те, що розвиток бездротових технологій зробили налаштування і впровадження бездротового інтерфейсу більш практичним і економічним, ситуація вибору бездротового інтерфейсу для СУРП ускладнюється наявністю широкого і заплутаного набору доступних протоколів, що розрізняються швидкістю передачі даних та діапазоном. Така ситуація ускладнює вибір конкретного протоколу в контексті СУРП. Далі порівнюються та підсумовуються найпопулярніші варіанти бездротового зв'язку для вбудованих систем.

2.3.1 Критерії оцінки бездротових комунікаційних протоколів

Комунікаційні протоколи - ті «мовні» правила, які дозволяють двом або більше комунікаційним системам передавати інформацію та «розмовляти» між собою - є повсюдними у сучасному світі. Вони пройшли довгий шлях, розростаючись у кількості і здатні дедалі більше ускладнюватись, оскільки перший протокол RFID був запропонований в 1948 році[14].

На сьогоднішній день комунікаційні протоколи стали настільки важливими, що їх управління, як правило, є колективна робота представників промислових організацій, які прагнуть стандартизувати передачу даних для задоволення потреб певних галузей чи ринків.

Організації, що розробляють стандарти (SDO), такі як Міжнародна організація зі стандартизації (ISO), Інститут інженерів електротехніки та електроніки (IEEE) та Проект партнерства третього покоління (3GPP) взяли на себе відповідальність за розробку, оновлення та обробку відповідних ринкових стандартів зв'язку. Інші бездротові протоколи ініційовані приватними підприємствами, наприклад протокол Z-Wave, розроблений Zensys для домашньої автоматизації.

Незалежно від їхнього походження, з нашим все більш пов'язаним громадським та приватним життям, всі види протоколів бездротового зв'язку набувають широкого застосування і мають розглядатися для СУРП.

2.3.2 Wi-Fi

Wi-Fi був створений для того, щоб побутові та офісні пристрої та обладнання могли бездротово підключатися та взаємодіяти з Інтернетом.

2.3.3 802.15 - Бездротові персональні мережі

Стандарт 802.15 був створений для взаємозв'язку персональних електронних пристроїв

2.3.4 802.15.1 - WPAN / Bluetooth

802.15.1 було введено, щоб дозволити обмін даними між фіксованими та мобільними пристроями на короткі відстані з використанням короткохвильових УВЧ-радіохвиль у промислових, наукових та медичних радіодіапазонах.

2.3.5 802.15.4

Стандарт 802.15 був створений, щоб запропонувати дешевий, низький рівень пропускної здатності даних, повсюдний зв'язок між пристроями, з малою або відсутністю базової інфраструктури.

2.3.6 Протоколи передачі стільникових даних

Протоколи стільникових даних були розроблені, щоб забезпечити загальноміське, національне та глобальне покриття та безперебійну мобільність від однієї точки доступу до іншої.

2.3.7 LoRa

LoRa здійснив революцію в IoT, забезпечивши передачу даних на великій відстані, використовуючи дуже мало енергії. Він був створений для пристроїв, які повинні працювати від батареї та тривати довгий час без підзарядки.

2.3.8 Sigfox

Sigfox був створений як програмне комунікаційне рішення для управління всією мережею та обчислювальної складності в хмарі, а не на пристроях, та різко зменшити споживання енергії та вартість підключених пристроїв.

2.3.9 Z-Wave

Z-Wave був представлений для забезпечення безпечної технології розумного будинку, яка не буде перешкоджати сигналу Wi-Fi і працюватиме на низькій потужності.

2.3.10 RFID та NFC

RFID був створений для ідентифікації інформації без прямої видимості, можливості оновлення інформації в режимі реального часу. NFC розроблений для автоматичного збору бездротових даних.

2.3.11 Діаграма результативності

Для швидкого ознайомлення нижче розглянемо візуалізацію роботи різних протоколів зв'язку. Таким чином ми порівняємо діапазон, швидкість передачі даних та енергоспоживання протоколів бездротового зв'язку.

Рисунок 2.2 нижче показує можливості кожного протоколу для зручного перегляду.

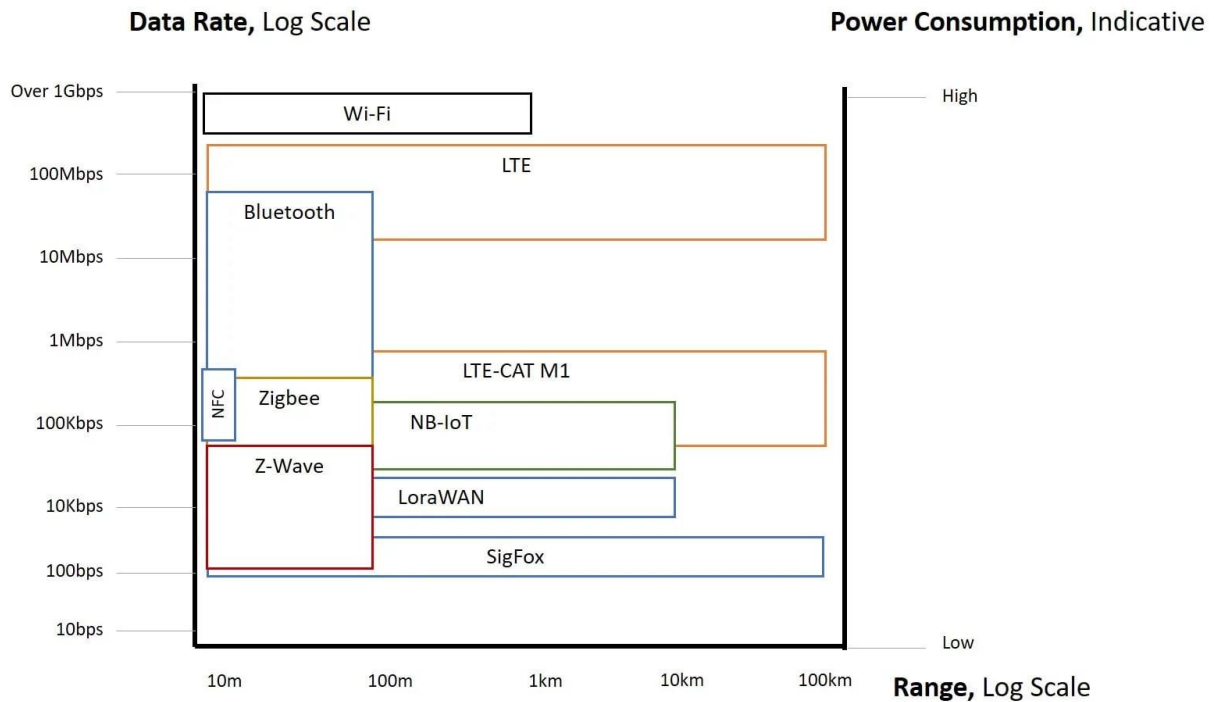


Рисунок 2.2 – Порівняння протоколів бездротового зв'язку.[14]

Як можна бачити з рисунка 2.2, між різними стандартами є чіткі ділянки, що перекриваються, а також області, де один з них може мати кращі результати.

Що стосується діапазонів, Sigfox, LoRa та 4G LTE є лідерами. Однак сильно урбанізовані райони створюють численні перешкоди для сигналів, зменшуючи ефективний діапазон для кожного в реальних реалізаціях. 4G і LTE використовують стільникові мережі, тому при розгляді цих протоколів необхідно враховувати щільність або обсяг пристроїв в одній стільниковій мережі.

Для домашньої автоматизації стандартні домашні умови означають, що кожен вузол повинен ефективно взаємодіяти з маршрутизатором або концентратором у конфігурації зіркової мережі. Таким чином, Zigbee або навіть Z-wave найкраще підходять для цього застосування. Однак у промислових установках важливим є варіант розширення діапазону для реалізації топології сітки або розширена зіркова конфігурація мережі. У порівнянні, для програм зв'язку M2M до 10 см NFC - найсильніший вибір.

Розглядаючи швидкість передачі даних, 802.11 та 4G LTE надають більше місця для надсилання інформації, включаючи такі програми, як моніторинг у реальному часі промислових систем або системи обробки відео інформації. Помірні швидкості передачі даних підходять для 802.11ah, Zigbee, Thread, Z-wave та NFC, тоді як низькі швидкості передачі даних віддають перевагу Sigfox.

Аналогічним чином, розгляд споживання енергії показує нам, що Wi-Fi потребує більше енергії, щоб функціонувати, але потрібно мати на увазі, що 802.11ah (Hallow) є багатообіцяючою альтернативою і порушує цю думку. Sigfox та LoRaWAN є найбільш корисними у програмах з низьким енергоспоживанням.

2.3.12 Перспективні комунікаційні протоколи

Стандарт 802.11 - і висока швидкість передачі даних, яку він забезпечує – стрімко набирає популярності для житлових та комерційних цілей. 802.15.4 та протоколи LR-WPAN забезпечують можливість підключення промислових вимірювань та засобів управління, а також застосувань в автоматизації будинків. 802.15.1 забезпечує засіб однорангового (P2P) зв'язку. А 802.15.4w LPWAN передає потужність близько 10 мВт, забезпечуючи при цьому зв'язок на великі відстані до 15 км у сільській місцевості, а також проникатиме до будівель у міській місцевості.

Крім того, нові протоколи бездротового зв'язку продовжують створюватися, і в останні роки деякі нові інновації, такі як протокол MIOTY (Behr), Wi-Fi низької потужності (InnoPhase), Sigfox, Multefire та багатовідомий 5G, почали набирати популярність на ринку.

2.3.13 Висновки

Враховуючи усі переваги та недоліки, було прийнято рішення використовувати протокол Wi-Fi для забезпечення бездротової комунікації СУРП. Проте слід визначити що радіозв'язок дозволяє досягти більшої дальності та

надійності. Додаткове дослідження переваг та недоліків різних протоколів радіозв'язку було проведене у роботі[11]. Цей вибір зумовлений наявністю вбудованого Wi-Fi адаптера у обраному ОПК, простотою використання існуючих веб-технологій, та достатньою швидкістю для передачі відеоданих.

Wi-Fi зв'язок у розробленому прототипі СУРП реалізований за рахунок вбудованого адаптера RTL8189FTV QFN24. Даний адаптер відрізняється високою якістю, і підтримується всіма операційними системами на базі Linux або Windows. Адаптер RTL8189FTV QFN24 може працювати в режимах IEEE 802.11b, IEEE 802.11g и IEEE 802.11n. Підключення до Wi-Fi автоматизовано за допомогою утиліти що виконується при запуску СУРП. Ця програма виконує пошук всіх доступних мереж та швидко підключається до встановленої, що є важливим для швидкого налаштування СУРП у небезпечних зонах.

Потрібно зазначити, що в умовах дослідження закритих приміщень радіозв'язок з СУРП не є надійним, тому у практичних реалізаціях програмно-апаратних інфраструктур для управління наземною рухомою платформою дуже часто використовується дротовий зв'язок. У даній роботі задля підвищення надійності розглядається використання обох видів зв'язку. Кількісне дослідження надійності системи буде проведене у наступних розділах.

2.4 Вибір відеокамери для системи управління рухомою платформою

Відеокамера є найважливішим пристроєм для забезпечення локалізації СУРП у просторі. Для реалізації задачі локалізації СУРП у просторі використовуються алгоритми візуальної одометрії, які можуть використовувати рішення як з стерео камерами, наприклад[13], так і монокулярним зором.

Монокулярний підхід використовують для спрощення СУРП, як наприклад у роботі[11], оскільки це дозволяє уникнути використання спеціалізованих стереокамер, а також для зниження вимог до обчислювальних ресурсів. Тим не менш, реалізації з стереокамерами дозволяють уникнути «масштабної

невизначеності», а також складних і ненадійних процедур ініціалізації глибини орієнтирів, що в системах монокулярної одометрії займають більшу частину ресурсів, ніж пошук орієнтирів чи обрахунок переміщення. Таким чином, використання стереокамери дозволяє підвищити надійність алгоритму візуальної одометрії, що є критичним у випадку втрати зв'язку з СУРП при дослідженні небезпечної зони.

Стереозір дозволяє за допомогою порівняння двох зображень визначити просторові координати двох точок, дозволяє працювати з більш абстрактними і примітивними орієнтирами – точками, і не потребує їх класифікації. В результаті стереосистема менш вибаглива до навколишнього середовища. Враховуючи описані фактори, у цій роботі для захоплення зображення і локалізації СУРП у просторі була використаний підхід з використанням стереобачення, що з'єднаний з мікрокомп'ютером OrangePiZeroPlus за допомогою протоколу USB.

У даній роботі була використана камера ELP-960P2CAM на базі двох сенсорів OV9750. Камера сумісна з усіма версіями Orange Pi. Вона прикріплюється до OrangePiZeroPlus через стандартний роз'єм USP 2.0.

Розміри камери: 80 x 16,5 мм. Розширення фото, отриманих з камери — 2560X960 пікселів. Підтримувані формати відео — 2560X960 @ 60fps/2560X720 @ 60fps /1280X480 @ 60fps /640X240 @ 60fps.

Обробка зображення використовується за допомогою обгортки на мові програмування Java, яка буде детально розглянута у наступному розділі.

2.5 Огляд шасі для мобільних платформ

Шасі для мобільних платформ складаються з структурних деталей та двигунів, і на ринку існує великий вибір готових варіантів. У більшості випадків структурні деталі виготовляються з акрилу, із вбудованими отворами для монтажу апаратної інфраструктури. Раніше був популярним варіант переробки готових продуктів для створення прототипів рухомих платформ, проте простота

роботи із сучасними варіантами готових шасі та їх низька ціна значно знижують привабливість даного варіанту.

Розглянемо різні типи шасі, які можна наразі знайти на ринку[15], їх переваги та недоліки.

Деякі набори шасі роботів мають складні методи руху, ніжки, гусениці та тризіркові колеса, щоб назвати декілька. Хоча це має наукову цінність, і може бути розглянуте у наступних роботах, у даній роботі не має необхідності у таких системах. Отже, було обрано цілком розумні, хоча і базові, колеса на версії двигуна.

Є комплекти з повним приводом і шестиколісним приводом. Вони можуть бути досить потужними і потребуватимуть більших контролерів двигуна. Вони також мають великі вимоги щодо енергоспоживання батареї, і збільшують ймовірність перевантаження. Подібні системи також ускладнює прокладення дротів.

Двоколісний привід є найпростіший для підключення. Для балансу зазвичай потрібно третє колесо. Це може бути коліщатко, роликовий м'яч або просто тефлонові ковзанки для малих роботів. Двома колесами також найпростіше керувати, уникаючи деяких проблем тертя, які спостерігаються у роботів, що використовують чотири або більше коліс. Два колеса не матимуть тягової сили чотирьох чи шестиколісних приводів, але вони прості і будуть працювати. Вони також дешевші.

У ході роботи були виявлені певні недоліки такої системи. Платформа має схильність проковзувати на нерівній поверхні, наприклад килимах. Хоча цей недолік є суттєвим для рухомих платформ дослідження території у небезпечних умовах, він не впливає на результати досліджень.

2.6 Вибір апаратних компонентів системи управління виконавчими механізмами рухомої платформи.

Найпростіші контролери двигунів просто забезпечують управління потужністю двигуна, що приводиться в дію безпосередньо від штифтів GPIO, наприклад L298[15]. Це найдешевше рішення: загальний контролер двигуна L298N можна підключити до деяких контактів вводу-виводу на ОПК. Вони досить надійні і легко доступні протягом тривалого часу. Вони гнучкі, але використання таких деталей займе більше місця, і їх потрібно підключати від точки до точки, додаючи складності в побудові.

Інші контролери ж складні, багато з яких приховують власний мікроконтролер, схожий на Arduino, разом із мікросхемами управління двигуном[15]. Незважаючи на те, що найдешевші та найгнучкіші способи - це найпростіші контролери, ті, що мають більш високу інтеграцію, зменшують розмір, збережуть низький рівень використання штифтів і можуть спростити систему. Вони також часто вбудовуються в блок живлення.

Контролери двигуна можна придбати як повністю інтегровані плати для певних моделей ОПК, плати, розроблені так, щоб точно розміститися зверху ОПК. Вони, як правило, мають високий рівень інтеграції, як вже обговорювалося раніше, але значно обмежують можливості системи. Оскільки ОПК OrangePiZeroPlus має небагато таких контролерів, і вони погано пристосовані до поставлених задач, було прийнято рішення їх не використовувати.

Враховуючі розглянуті параметри, було прийнято рішення створити власний контролер двигунів на основі модуля L298N та мікроконтролера STM32F401.

2.6.1 STM32F401 як мікроконтролер для управління виконавчими механізмами в режимі реального часу

Реалізація вимоги точного контролю руху потребує контролю параметрів двигунів в реальному часі. Для вирішення цієї задачі використання мікрокомп'ютера OrangePiZeroPlus не є оптимальним. Тому було прийнято рішення використовувати апаратну інфраструктуру що є поєднанням мікрокомп'ютера OrangePiZeroPlus та мікроконтролера STM32F401. STM32F401 є надійною та дешевою платформою для виконання обчислень у реальному часі, з великим обсягом документації та стандартизованими рішеннями для керування робото-технічними системами, що робить дану систему оптимальною для управління виконавчими механізмами СУРП.

Струм, необхідний для живлення STM32F401 є вкрай незначним у порівнянні з вимогами стратегічного рівня обчислень, тому було прийнято рішення підключити живлення мікроконтролера до виводу струму на пінах GPIO мікрокомп'ютера OrangePiZeroPlus. Обмін інформацією здійснюється за допомогою протоколу UART.

Мікроконтролер STM32F401 формує керівні ШИМ-сигнали для двох двигунів, що подаються на керівний міст L9110 H-bridge, що дозволяє контролювати високу напругу двигунів. З іншої сторони, мікроконтролер STM32F401 на вхід отримує сигнали від оптичних датчиків перешкоди, встановлених на кожному двигуні. Це дозволяє з високою точністю контролювати швидкість обертання кожного з двигунів, і відповідно, за допомогою програмних ПД-регуляторів вимірювати і контролювати зміни положення мобільної платформи.

2.6.2 Використання контролера L298N H-bridge для управління двигунами системи управління рухомою платформою

Модуль L298N H-bridge, призначений для керування двигунами постійного струму за допомогою низької напруги на виході від GPIO[16]. За рахунок того, що струм, що іде на двигуни, і струм, який ними керує, є незалежними, цей модуль дозволяє керувати двигунами постійного струму через мікроконтролер STM32F103, потужності якого не вистачає для живлення двигунів.

Модуль L298N H-bridge може підключатися до двох двигунів і керувати ними одночасно. На чотири входи підключаються виходи двигунів, на виходи – чотири керуючі ШИМ-сигнали для двигунів, які ідуть на I/O контакти GPIO. Також модуль потребує підключення стороннього джерела струму, яке використовується для живлення двигунів.

Для керування кожним двигуном постійного струму використовуються два цифрові входи управління. Перший дозволяє обрати напрямок обертання двигуна, у той час як другий цифровий вхід керує швидкістю обертання за допомогою широтноімпульсної модуляції(ШИМ). Результат подачі сигналів можна визначити за допомогою таблиці істинності, зображеної у табл. 2.1.

Варто звернути увагу, що напрямок обертання двигуна залежить не лише від фактичного сигналу, а і від способу монтування двигунів. Напрямок двигуна завжди можна змінити шляхом підключення його проводки у іншому напрямку.

Таблиця 2.1 — L298N Таблиця істинності[16]

Вхід		Вихід		
IA	IB	OA	OB	Опис
0	0	0	0	Вимкнений
1	0	1	0	Вперед
0	1	0	1	Назад
1	1	1	1	Вимкнений

L298N містить два канали управління двигунами, наявність яких означає, що керування двома двигунами може відбуватись незалежно один від одного. Для кожного каналу двигуна можна використовувати таблицю істинності 2.1.

Розглянемо усі з'єднання за допомогою таблиці 2.2.

Таблиця 2.2 — Таблиця з'єднань[16]

Вхід	Опис
B-IA	Двигун В вхід А
B-IB	Двигун В вхід В
GND	Заземлення
VCC	Операційна напруга 2.5-12В
A-IA	Двигун А вхід А
A-IB	Двигун А вхід В

Як можна побачити на таблиці 2.2, для управління швидкістю обертання слід використовувати вхід IA і вихід IB для керування напрямком обертання.

2.7 Джерело живлення

2.7.1 Види джерел живлення, що використовуються в робототехніці

Існує велика кількість типів акумуляторів[12], однак для простоти їх можна поділити на дві групи – батареї що підходять для роботів, та батареї що не придатні для робототехніки.

Акумулятори що підходять для робототехніки:

- Li-Ion літій-іонні батареї;
- Li-Poly літій-полімерні батареї;
- NiMH — нікель-металогідридні батареї.

Розглянемо акумулятори які не придатні для робототехніки, та причини такого становища.

Свинцево-кислотні акумулятори, такі як VRLA, SLA, гель або AGM, не пристосовані до частих циклів зарядки-розрядки і зазвичай використовуються як резервний блок живлення для стаціонарних установ. Ще одним чинником, через який свинцево-кислотні акумулятори не використовуються в робототехніці, є їх велика вага.

NiCd, або нікель-кадмієві батареї багато у чому схожі на нікель-металогідридні батареї, але мають значний недолік перед ними через використання токсичних металів і не мають переваг.

NiH₂, або нікель-водневий акумулятори, хоча і мають дуже високі характеристики, потребують рідких матеріалів та мають високу ціну. Тому нікель-водневий акумулятори не використовуються в робототехніці.

Інші типи є або більш дорогими, або складними у покупці та зарядці, тому не розглядаються

2.7.2 Порівняння Li-Ion, Li-Poly і NiMH

Батареї Li-Poly дуже схожі за своїми характеристиками до літій-іонних акумуляторів. Можна сказати, що вони є підгрупою Li-Ion батарей. Їх особливість полягає в тому, що виробничий процес літій-іонних акумуляторів влаштований таким чином, що під час виробництва клітини Li-Ion повинні бути закріплені всередині металевого корпусу для збереження структурної цілісності. Li-Poly акумулятори, розроблені пізніше, відрізняються за своєю конструкцією та можуть утримувати себе без підтримки зовнішнього циліндра. У той же час електричні параметри Li-Ion і Li-Poly мало відрізняються.

Нікель-металогідридні батареї мають певні переваги, такі як низький внутрішній опір і хорошу потужність. У порівнянні з літій-іонними акумуляторами, вони є значно безпечнішими. Випадки вибухів нікель-металогідридних батарей є дуже рідкими. Проте співвідношення енергії до ваги є значно гіршими у порівнянні з літійовими акумуляторами.

2.7.3 Аналіз та обґрунтування обраної системи живлення.

Мобільна платформа дослідження території рухається за допомогою двох двигунів постійного струму, що рухають два колеса. Швидкість не є важливим параметром для мобільної платформи дослідження території, оскільки захоплення та обробка відеозображення, та подальше обчислення і передача даних займає досить великий час. Тому не має потреби у великій швидкості.

У даному випадку на таблиці 1.3 проаналізовані вимоги потужності для усіх систем рухомої платформи. Струм не приймається до уваги, оскільки у систем використовується багато пристроїв з різними вимогами до вхідного струму – деякі потребують високу напругу і низький струм, а інші – низьку напругу і високий струм. Використовуючи перетворювачі постійного струму, ми можемо використати 10В як мінімальну напругу для батареї, і розраховувати середній струм виходячи з цього.

Таблиця 2.3 Розрахунки потужності та струму

Компонент	Середня потужність (W)	Максимальна потужність(W)
Електромотор	6,2	36
Електромотор	6,2	36
Біноклярна камера	2	1
Модуль L298N	0,8	0,8
OrangePiZeroPlus	4	7
STM32F103C8T6	0,1	0,12
Wi-Fi модуль	0,7	0,9
Загальна потужність (W):	19	81,8
Загальний струм при 12V (W):	1,6	6,8

Отже, ми можемо сформулювати вимоги до нашої батареї:

- розмір має бути нижче 150 см³;
- вага має бути нижче 500г;
- номінальна напруга — близько 10–12В;
- максимальний струм розряду — не менше 9.3А;
- ємність — середній струм становить 2.1А, а мінімальний робочий час — 120 хвилин, тому необхідна потужність вище 4200mAh;
- низький струм саморозряду є перевагою.

Нікель-металогідридні батареї не будуть використовуватися, оскільки не містять достатньо енергії. У цьому випадку літієві батареї є кращими.

Li-Ion: акумуляторна батарея 3x18650 (3S2P) виготовлена з INR18650–35E Samsung 3500mAh.

- розміри: 67 x 57 x 19 мм;
- ном. напруга: 10,8 В (3,6 В на комірку);
- мін. напруга: 9В;
- макс. струм розряду: 8А;
- об'єм: 72,5 см³;
- вага: 150г;
- місткість: 4500mAh;
- ціна: близько \$ 15.

Цей варіант виглядав найбільш оптимально у ході аналізу. Питома енергія(відношення потужності до ваги) є набагато більшою за необхідну у даній предметній області. Ціна низька.

Li-Poly: акумуляторна батарея 3S1P виготовлена з LP616594 4700mAh 3.7V:

- розміри: 94 x 65 x 19 мм;
- ном. напруга: 10,8 В (3,6 В на комірку);
- мін. напруга: 9В;
- макс. струм розряду: 9.5А;
- об'єм: 113 см³;

- вага: 236г;
- місткість: 4700mAh;
- ціна: близько \$ 60.

Із переваг даної системи варто зазначити меншу вагу та розміри, проте питома енергія та ціна є набагато більшими. Дана система є прикладом низькотемпературного типу Li-Ion — максимальний струм — лише 2С, але цього достатньо для предметної області.

Подальший аналіз систем живлення показав необхідність розділу живлення моторів та обчислювальної техніки на два елементи живлення. Причиною даного рішення було те, що перетворювач постійного струму видавав максимальний струм у 2А, а цього не достатньо для роботи системи при старті двигунів.

Було прийнято рішення використовувати касету для 4 батарейок типу AA для живлення двигунів. Вона видає на виході загальну напругу 6В.

Для живлення ОПК та мікроконтролера використовується акумулятор Battery Charge Shield Lithium 18650 V3. Він зображений на рисунку 2.6. Battery Charge Shield Lithium 18650 V3 надає напругу 5В і підключається до OrangePiZeroPlus за допомогою інтерфейсу microUSB. Модуль автономного живлення є комбінацією двох пристроїв – зарядного пристрою для Li-Ion акумулятора та перетворювача напруги з 3,6В в 3 або 5В. Такі напруги дозволяють використовувати даний пристрій для живлення таких пристроїв як мікроконтролери Arduino, ОПК Orange Pi і Raspberry Pi і ін, а також зарядки мобільних пристроїв, виступаючи в якості автономного джерела живлення.

Для перемикання режиму роботи використовується малогабаритний перемикач. Для підключення додаткових споживачів передбачені три пари контактів з напругами 3В і 5В.

5V від блоку живлення основного модулю, створеного на основі літій-іонного акумулятора стандарту 18650.

Система навігації, камера та датчики під'єднані до основного керівного модуля за допомогою протоколу USB 2.0. Це дозволяє використовувати встановлені драйвери операційної системи для розробки програмної частини СУРП, спрощуючи розробку системи.

Для досягнення більшої відмовостійкості використовуються дротовий та бездротовий зв'язок одночасно. Дротова система має трохи більшу стійкість до відмову, тому є основною. Дротовий зв'язок відбувається використовуючи інтерфейс RS 485, що забезпечує надійне та дешеве з'єднання на далекій відстані. Бездротовий зв'язок на основі протоколу Wi-Fi використовується у разі втрати основного зв'язку або у ситуаціях де його використання не є можливим.

Оскільки управління певними виконавчими механізмами має відбуватися в реальному часі, було прийнято рішення використовувати окремий модуль виконавчих механізмів, реалізований за допомогою мікроконтролера STM32F401.

Модуль виконавчих механізмів передає чотири сигнали управління двигунами до системи живлення і управління двигунами – сигнали напрямку та потужності для кожного з двох двигунів.

Ці сигнали використовуються для модуляції струму що надходить з блоку живлення двигунів, створеного на основі чотирьох гальванічних елементів живлення стандарту AA.

Модуль виконавчих елементів також подає живлення та отримує інформацію від одометрів кожного з двигунів, вимірюючи їх швидкість що використовується для контролю двигунів в реальному часі.

2.9 Розробка функціональної схеми системи управління рухомою платформою

Для опису роботи основного керівного модулю та модулю виконавчих механізмів створимо функціональну схему СУРП, зображену на рисунку 2.4.

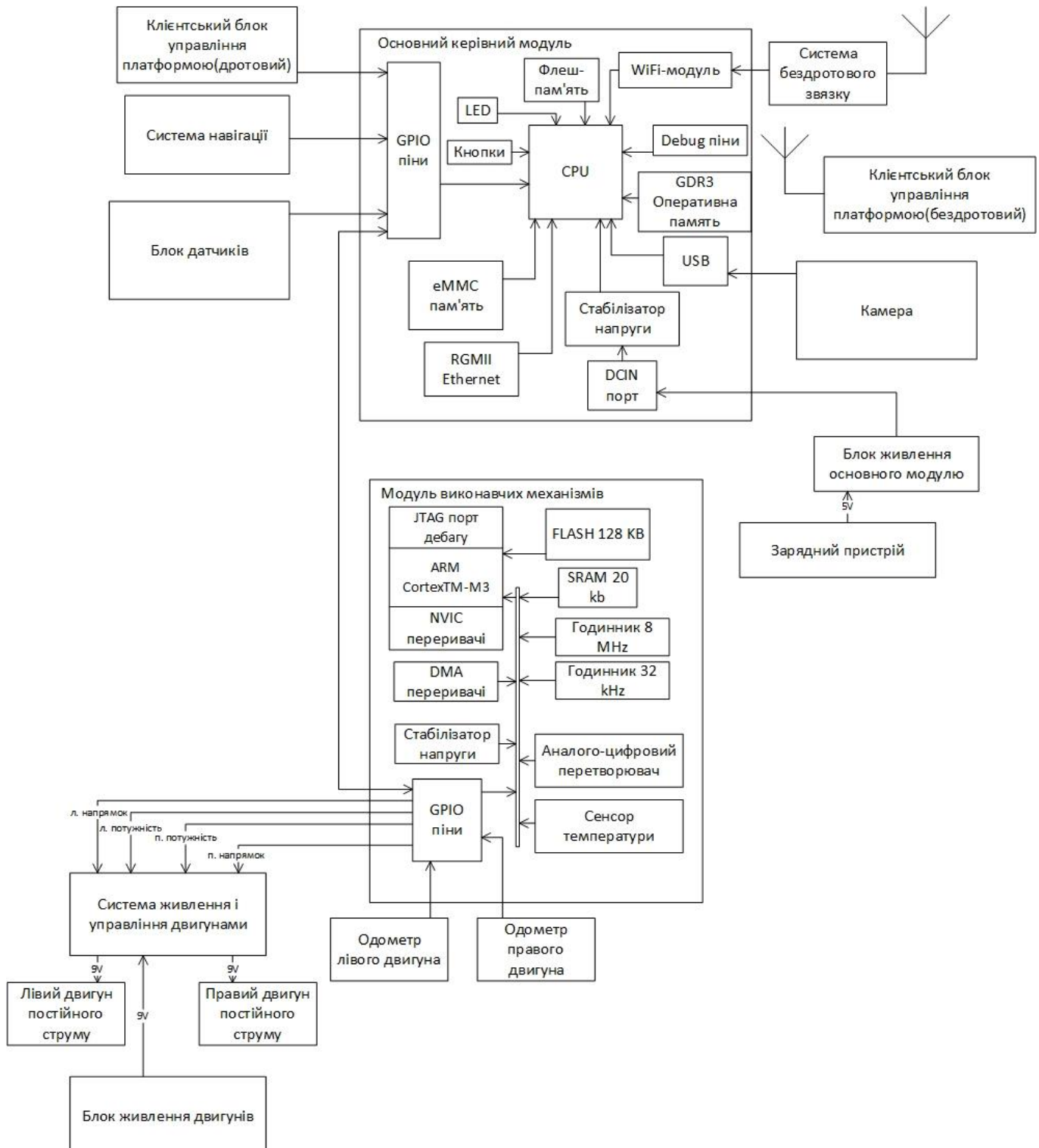


Рисунок 2.4 – Функціональна схема системи управління рухомою платформою.

Процесор ARM Cortex-M3 - це останнє покоління процесорів ARM для вбудованих систем[17]. Він є найкращим варіантом для використання у модулі виконавчих елементів СУРП, оскільки являє собою недорогу платформу, яка відповідає потребам впровадження мікроконтролерів, зі зменшеною кількістю виводів та низьким енергоспоживанням, забезпечуючи при цьому видатні обчислювальні характеристики та вдосконалену системну реакцію на переривання.

32-розрядний процесор RISC ARM Cortex-M3 відрізняється винятковою ефективністю коду, забезпечуючи високу продуктивність, яка очікується від ядра ARM, в обсязі пам'яті, як правило, пов'язаному з 8 та 16 бітними пристроями. Висока продуктивність дозволяє зменшити вимоги до програмного забезпечення, спрощуючи його розробку.

Родина STM32F103xx має вбудоване ядро ARM, тому сумісна з усіма інструментами та програмним забезпеченням ARM які використовуються в побудові СУРП.

Для зберігання програм і даних доступно до 128 Кбайт вбудованої Flash-пам'яті. Модуль виконавчих механізмів також містить 20 Кбайт вбудованої операційної пам'яті, доступних (читання / запис) з тактовою частотою процесора.

Для зчитування інформації з одометрів двигунів СУРП використовуються зовнішні контролери переривань / подій (EXTI). EXTI складається з 19 рядків детекторів краю, що використовуються для генерації запитів переривання або події. Кожен рядок може бути незалежно налаштований на вибір події спрацьовування (висхідний край, спадаючий край, обидва). Очікуючий реєстр, зберігає статус запитів на переривання. EXTI може виявити зовнішню лінію з шириною імпульсу, меншою за внутрішній тактовий період APB2.

Вибір системного тактового сигналу виконується під час запуску, однак внутрішній генератор RC 8 МГц вибирається як тактовий сигнал центрального процесора за замовчуванням. Можна вибрати зовнішній тактовий сигнал 4-16 МГц, який контролюється на предмет відмов. У випадку відмови він вимикається,

і слід переривання програмного забезпечення. Подібним чином, при необхідності доступне повне управління перериваннями годинника PLL (наприклад, при відмові зовнішнього генератора). Враховуючи необхідність керування двигунами СУРП у реальному часі, і низьку надійність тактового сигналу основного керівного модулю, було прийнято рішення використовувати внутрішній генератор RC 8 МГц як тактовий сигнал центрального процесора.

Модуль виконавчих механізмів має вбудовану схему скидання живлення (PDR). Цей модуль завжди активний і забезпечує належне живлення в межах 2 В. Пристрій має вбудований програмований детектор напруги (ПДН), який контролює джерело живлення і порівнює його з пороговим значенням. Переривання може бути генеровано, коли вхідний струм є нижче чи вище порогу. Потім процедура переривання може генерувати попереджувальне повідомлення та / або перевести модуль виконавчих механізмів в безпечний стан. ПДН вимкається програмною інфраструктурою СУРП і забезпечує цілісність модуля виконавчих механізмів у випадку відмови блоку живлення.

7-канальна система загального призначення прямого доступу до пам'яті(ПДП) здатна управляти передачею інформації з пам'яті в пам'ять, з виконавчих механізмів в пам'ять та з пам'яті у виконавчі механізми. Контролер ПДП підтримує кругове управління буфером, уникаючи генерації переривань, коли контролер досягає кінця буфера. Кожен канал підключений до виділених апаратних запитів ПДП з підтримкою програмного запуску на кожному каналі. Конфігурація здійснюється за допомогою програмного забезпечення, і розміри передачі між джерелом і пунктом призначення не залежать. У контексті роботи модуля виконавчих механізмів СУРП ПДП використовується для забезпечення з'єднання з основним керівним модулем за допомогою протоколу USART, а також для керування таймерами управління TIMx що забезпечують вимірювання швидкості.

Модуль виконавчих механізмів містить 3 синхронізованих стандартних таймери. Ці таймери засновані на 16-розрядному лічильнику автоматичного

перезавантаження вгору / вниз, та мають 4 незалежних канали для вхідного / вихідного порівняння, ШІМ або один вихід в імпульсному режимі. Це надає можливість використовувати до 12 вхідних захоплень / вихідних порівнянь / ШІМ. Вони можуть працювати разом з розширеним таймером управління за допомогою функції Timer Link для синхронізації або ланцюжка подій. Лічильник можна заморозити в режимі налагодження. Синхронізовані стандартні таймери використовуються у модулі виконавчих механізмів для формування ШІМ сигналу для управління двигунами мобільної платформи.

Вбудований інтерфейс USART здатний спілкуватися зі швидкістю до 4,5 Мбіт / с. Інші доступні інтерфейси взаємодіють зі швидкістю до 2,25 Мбіт / с. Вони забезпечують апаратне управління сигналами, відповідають стандарту ISO 7816 та мають функцію LIN Master / Slave. Всі інтерфейси USART можуть обслуговуватися контролером ПДП. Висока швидкість та простота у використанні були причинами вибору USART як інтерфейсу для комунікації між основним модулем керування та

2.10 Висновки

У цьому розділі були розглянуті існуючі варіанти компонентів для створення рухомої платформи. Були розглянуті причини вибору одноплатного мікрокомп'ютера для виконання обчислень, переваги та недоліки існуючих ОПК, і обґрунтовані причини вибору OrangePiZeroPlus. Були розглянуті існуючі протоколи бездротового зв'язку, обґрунтовані причини вибору системи живлення, а також розглянуто вибір відеокамери, шасі та системи управління двигунами. Було обґрунтовано чому вибір саме цих компонентів дозволяє виконати швидко, надійну та дешеву збірку прототипу для виконання досліджень.

3 ФРЕЙМВОРК SPRING ЯК ОСНОВА ПРОГРАМНОЇ ЧАСТИНИ СИСТЕМИ УПРАВЛІННЯ РУХОМОЮ ПЛАТФОРМОЮ

3.1 Фреймворк Spring як основа системи управління рухомою платформою

Spring Boot - це мікрофреймворк з відкритим кодом, який використовується для побудови програм Spring за допомогою мікросервісів. Створений Pivotal Software, Inc., він може використовуватися як для традиційного розгортання WAR, так і для окремих програм Java. Spring Boot допомагає розробникам негайно розпочати кодування, не витрачаючи час на підготовку та налаштування середовища. На відміну від інших фреймворків Java, він забезпечує гнучкі конфігурації XML, надійну пакетну обробку, транзакції баз даних, простий робочий процес, а також широкий спектр інструментів для розробки.

3.2 Переваги фреймворка Spring як програмної основи системи управління рухомою платформою

Доцільність використання фреймворку Spring як основи програмної частини системи управління рухомою платформою полягає в тому, що цей фреймворк має вбудовані компоненти для розробки сервісів управління, інтеграції сервісів управління в єдину систему і тестування[18]. Це дозволяє з найменшими витратами розробити основні програмні сервіси системи управління рухомої платформи. Під час розробки програмної частини рухомої платформи дотримуємося стратегії "Використання готових конфігурацій за замовчуванням". У цьому випадку є доцільним використання шаблонних та інших конфігурацій, у якості основи програмної частини системи управління рухомою платформою.

Наступною перевагою фреймворка Spring у контексті використання для СУРП полягає в тому, що він базується на мові програмування Java. Ця мова

програмування є однією з найпопулярніших мов програмування для розробки вбудованих систем.

Фреймворк Spring має величезну спільноту користувачів. Це дозволить швидко знайти розробників(програмістів) для створення програмної частини СУРП.

Система управління рухомою платформою передбачає наявність великої кількості різноманітних функцій. До них відноситься наступні функції:

- управління напрямком руху;
- забезпечення зв'язку оператора і СУРП у процесі управління;
- передача і обробка відеоданих.

У цьому випадку доцільно використання багатопоточності, що забезпечує фреймворк Spring. Наприклад, коли основний потік виконує функції управління рухом, інші паралельні потоки можуть виконувати множину функцій СУРП, перелічені вище.

Деякі додаткові переваги включають:

- зменшує час, витрачений на розробку, і підвищує загальну ефективність команди розробників;
- допомагає автоматично налаштувати всі компоненти для виробничого додатка Spring;
- сприяє створенню та тестуванню програм на основі Java, забезпечуючи налаштування за замовчуванням для модульних та інтеграційних тестів;
- допомагає уникнути всілякої ручної роботи з написання зразкового коду, анотацій та складних конфігурацій XML;
- поставляється із вбудованими HTTP-серверами, такими як Jetty та Tomcat, для тестування веб-додатків;
- інтеграція Spring Boot з екосистемою Spring, яка включає Spring Data, Spring Security, Spring ORM та Spring JDBC, є простою;

- надає безліч плагінів, які розробники можуть використовувати для плавної та легкої роботи із вбудованими базами даних;
- дозволяє легко підключатись до служб баз даних, таких як Oracle, PostgreSQL, MySQL, MongoDB, Redis, Solr, ElasticSearch, Rabbit MQ, ActiveMQ та багатьох інших;
- забезпечує підтримку адміністратора - ви можете керувати через віддалений доступ до програми;
- поставляється з вбудованим контейнером для сервлетів.
- пропонує гнучкість у налаштуванні конфігурацій XML, Java Beans та транзакцій баз даних;
- пропонує легкий доступ до інтерфейсу командного рядка, що робить розробку та тестування програм Spring Boot, побудованих за допомогою Java або Groovy, гнучкими.

3.3 Недоліки фреймворку Spring як фактори що обмежують програмну функціональність фреймворку у складі системи управління

Першим недоліком використання фреймворку Spring як основи програмної частини СУРП, є складність роботи з системою контролю версій. Отже, це дещо ускладнює процес супроводження програмної частини системи управління рухомою платформою.

Внутрішня архітектура фреймворку вимагає встановлення багатьох додаткових залежностей (що іноді залишаються невикористаними), збільшується розмір файлу розгортання. Але, сучасні мікрокомп'ютери мають значні обчислювальні потужності, і, певною мірою, можуть компенсувати цей недолік. У цьому випадку вартість апаратної частини СУРП може значно збільшуватися.

Артефакт Spring Boot можна запускати безпосередньо в контейнерах Docker. Це корисно отримати, коли потрібно швидко створити мікросервіси. Однак деякі розробники стверджують, що оскільки Spring Boot був

спроектований для того, щоб бути легким та рухливим, його не слід використовувати для монолітних додатків.

Хоча Spring Boot постачається з деякими основними інструментами для журналів та моніторингу стану вашого додатка, цього недостатньо. Такі інструменти, як Retrace, допомагають командам контролювати програми Java. Цей інструмент допомагає виявляти повільні запити SQL, забезпечує звіти про продуктивність та використання процесора та відображає найпоширеніші помилки, інтерпретуючи журнали.

До того ж оновити застарілий код Spring може бути досить складно. Можна подолати цю проблему за допомогою таких інструментів, як Spring Boot CLI (інтерфейс командного рядка), які допоможуть перетворити застарілий код.

Необхідно розуміти багато основних систем Spring (і трохи історії Spring також), а також деякі розширені теми, щоб змінити та усунути несправності у роботі.

3.4 Система передачі відеоінформації користувачу рухомої платформи на основі Video4Linux4Java

Важливим завданням рухомої платформи є дослідження території і передача відеоінформації користувачу. Існує багато програмних систем, що дають можливість захоплювати, обробляти і передавати зображення з використанням протоколу TCP/IP.

Оскільки основою фреймворку для побудови СУРП є мова програмування JAVA для захоплення зображення доцільно використати бібліотеку Video4Linux4Java (v4l4j)[19]. Оскільки вона розроблена на мові програмування JAVA, ця бібліотека легко інтегрується у СУРП на основі Spring.

Ця бібліотека реалізована у вигляді Java-пакету, що надає доступ до інтерфейсу захоплення відео API Video4Linux (V4L). У результаті дослідно-конструкторської роботи було порівняно декілька бібліотек роботи з відеоданими,

і було виявлено що бібліотека v4l4j найбільш стабільно і швидко виконує поставлене завдання. Дані у форматі BufferedImage, що оптимізований для швидкої трансформації в інші формати, надаються через Webcam Capture. Для збереження, аналізу та виводу зображення у даній роботі використовується формат jpeg. Бібліотека v4l4j надає наступний функціонал:

- отримувати інформацію про відеопристрій, наприклад, кількість і тип відеовходів, підтримувані формати зображень, роздільну здатність і стандарти відео та доступні тюнери;
- кадри захоплення з будь-яких пристроїв із підтримкою V4L, включаючи USB / Firewire Веб-камери, карти захоплення відео та карти тюнерів;
- доступ до елементів керування відео, таких як яскравість, контрастність, коефіцієнт посилення, нахил, фокус;
- тюнери доступу та керування (частота отримання / встановлення, доступ до потужності прийнятого сигналу).

3.5 Інтегрована база даних для збереження інформації рухомої платформи

Будь-яка наземна система для дослідження території передбачає збереження зібраної інформації з датчиків, сенсорів, відеокамери, системи навігації, команд управління на борту рухомої платформи. Таким чином в СУРП має бути інтегрована база даних.

Для подальшого використання відео дані зберігаються. У ході дослідно-конструкторської роботи зображення зберігалися локально, проте використання хмарних файлоховищ також можливе. Посилання на зображення, разом з метаданими зберігаються у локальній Sqlite[20] реляційній базі даних.

Варто бути ретельним у виборі платформи управління базами даних. використовувати. Зрештою, неякісний вибір може ускладнити оптимізацію цієї частини роботи або інших ключових завдань.

SQLite це загальнодоступна система управління реляційними базами даних (СУБД), що означає, що вона може використовуватись безкоштовно. Це також легка, зручна і крос-платформна система, не кажучи вже про ефективне завантаження необхідних даних.

SQL або мова структурованих запитів - це мова, яка в основному використовується при програмуванні. Зокрема, ця мова дозволяє отримати доступ до всіх даних, що зберігаються в табличній СУБД, та керувати ними.

SQLite знаходиться у відкритому доступі, це означає, що існує можливість вільно використовувати його код як комерційно, так і приватно. Це гарантує відсутність юридичних обмежень щодо використання, модифікації та навіть розповсюдження платформи.

На відміну від опцій клієнта / сервера SQL, таких як MySQL, SQLite оптимізований для економії та простоти та вимагає значно меншої конфігурації. У цьому контексті він не призначений конкурувати з рішеннями клієнт / сервер.

Натомість він більше зосереджений на забезпеченні локального сховища даних і може бути ефективним вибором для малих та середніх проєктів. Більше того, це популярний механізм баз даних для різноманітних платформ, таких як вбудовані системи (на зразок тих, що використовуються в смартфонах), операційні системи (ОС) та браузері.

SQLite це легкий та автономний RDBMS, упакований у дуже малий розмір.

Насправді це файловий і безсерверний сервер. Інші СУБД, такі як MySQL, потребують окремого сервера, щоб насправді функціонувати, також відомого як архітектура клієнт / сервер.

Залежно від системи, в якій він встановлений, SQLite може займати лише близько 600 КБ місця. Оскільки він також автономний, не має необхідності мати справу з встановленням будь-яких залежностей.

Об'єкти, що складають бази даних SQLite, включаючи таблиці, схему та індекси, централізовані лише в одному файлі ОС.

Все це може прирівнюватися до надійних характеристик, не кажучи вже про високий рівень надійності та загальної стабільності. Зокрема, SQLite може працювати в 10–20 разів швидше, ніж PostgreSQL, і вдвічі швидше, ніж MySQL. Різниця найбільш помітна для веб-сайтів із низьким навантаженням.

З самого початку, SQLite має нульову конфігурацію і не має мінімальних вимог. Він не включає конфігураційні файли і є безсерверним.

Це означає, що він не має жодного процесу встановлення. По суті, потрібно лише завантажити необхідні бібліотеки SQLite, щоб розпочати створення бази даних. Навіть це прирівнюється до невеликого розміру, який не створює значного навантаження на пам'ять комп'ютера.

Як СУБД, SQLite забезпечує широкі крос-платформні функціональні можливості та портативність.

Зокрема, існує можливість копіювати та передавати між 32 та 64-розрядними ОС. По суті, SQLite може працювати та компілюватися в різних системах, включаючи Windows, Linux та Mac OS.

Більше того, оскільки база даних SQLite зберігається лише в одному файлі, а не в колекції окремих, вона значно гнучкіша з точки зору її переносимості. Це ще більше розширює діапазон обчислювальних систем, які ця СУБД може охоплювати.

3.6 Діаграма компонентів системи управління рухомої платформи на основі фреймворку Spring

На основі результатів, отриманих у п 3.1, 3.2, і 3.3 була розроблена діаграма компонентів СУРП

Задля кращого опису результатів дослідження була розроблена і підтверджена результатами дослідно конструкторської роботи діаграма компонентів. На цій діаграмі(рис 3.1) виділені компоненти мобільної платформи

що показують базові елементи систем передачі і обробки команд керування, обробки відеоданих, а також даних швидкості двигунів.

Розглянемо діаграму компонентів більш докладно. На ній виділені базові елементи систем передачі і обробки команд керування, обробки відеоданих, а також обробки даних швидкості двигунів. Зображення з стереоскопічної відеокамери передаються через інтерфейс USB. Бібліотека Video4Linux4Java використовується для захоплення зображень. Інформація обробляється у Orb Slam 2, і виводить дані про місцезнаходження робота. Ця інформація, разом з відеоданими, зберігається у базі даних чи відправляється оператору через веб інтерфейс, залежно від функції що виконується у даний момент. Відповідно до сигналів оператора, додаток donkeybug.jar формує команди руху, і надає їх до мікроконтролера через інтерфейс USART. Мікроконтролер містить додаток receiver.c, що використовує три бібліотеки – GPIO.h, stateMachine.h, та encoder.h.

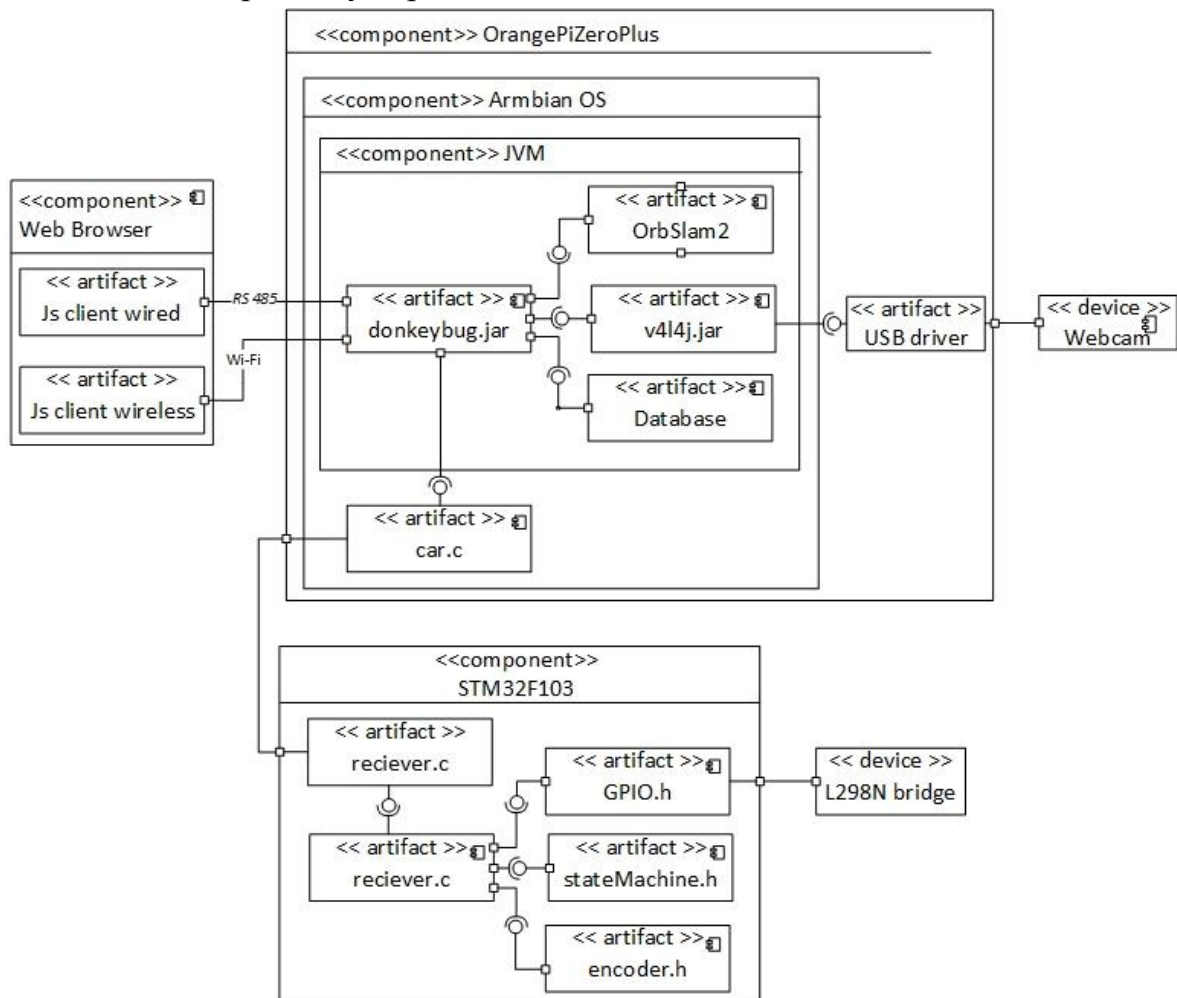


Рисунок 3.1 – Діаграма компонентів

GPIO.h надає зручний інтерфейс роботи з GPIO каналами для передачі команд керування двигунами. stateMachine.h є бібліотекою що містить логіку роботи кінцевого автомату управління двигунами. У той же час encoder.h є бібліотекою що дозволяє працювати з оптичним одометром, який вимірює швидкість роботи двигунів.

3.7 Висновки

Була досліджена і обгрунтована доцільність використання фреймворку Spring як основи програмної частини системи управління рухомою платформою. Були розглянуті переваги та недоліки фреймворку Spring. Розглянуто Video4Linux4Java як основу системи передачі відеоінформації користувачу рухомої платформи. Була досліджена та обгрунтована необхідність використання інтегрованої бази даних на основі Sqlite у складі СУРП. Була побудована діаграма компонентів СУРП на основі фреймворку Spring.

4 ПРОГРАМНА ІНФРАСТРУКТУРА СИСТЕМИ УПРАВЛІННЯ НАЗЕМНОЮ РУХОМОЮ ПЛАТФОРМОЮ

На рисунку 4.1 зображена Use Case діаграма СУРП. Вона складається з кейсів основного керівного модуля, виконавчого механізму і бази даних.

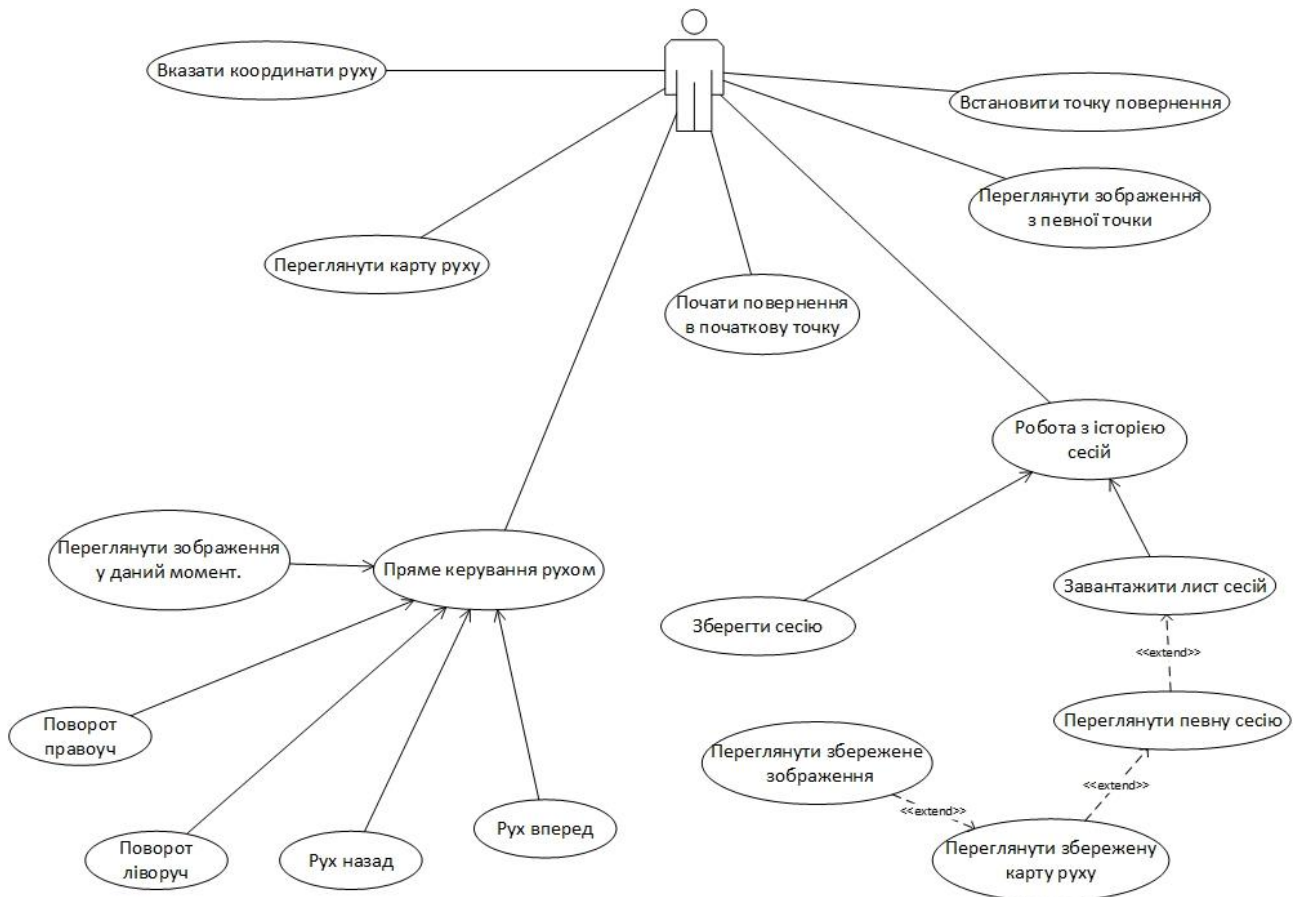


Рисунок 4.1 – Use Case діаграма рухомої платформи

Опишемо наведену діаграму більш детально. Першою основною функціональною категорією є основний керівний модуль, що відповідає

Для типової команди керування двигунами побудуємо таблиці із наступними полями: назва діяльності, її унікальний ідентифікатор, опис, актор, тригер спрацювання, попередні умови для виконання, умови після виконання, основний та альтернативний потоки розвитку дій та можливі помилки.

4.1 Сценарії використання системи

Використання візуальної одометрії дозволяє автоматизувати керування рухомою платформою, зменшуючи вірогідність людської помилки. Використовуючи зображення з камери створюється карта руху, на якій зображена траєкторія руху, а також перешкоди які були визначені системою. Користувач може дати команду рухатися у певну точку на цій карті, переглянути точку зору з будь-якої з минулих точок руху. Також користувач може надати команду повернутися у початкову точку. Та ж сама команда буде виконана автоматично якщо усі способи зв'язку з користувачем будуть втрачені. Звичайно, користувач може встановити будь-яку точку на карті як точку повернення.

Для типової команди руху в певну точку побудуємо таблиці із наступними полями: назва діяльності, її унікальний ідентифікатор, опис, актор, тригер спрацьовування, попередні умови для виконання, умови після виконання, основний та альтернативний потоки розвитку дій та можливі помилки.

Розглянемо діяльність «Set coordinates». Її детальний опис наведено у таблиці 4.1.

Таблиця 4.1 – Опис діяльності «Set coordinates»

Найменування	Set coordinates
ID	C_01
Передумови	Користувач розпочав сесію керування та отримав підтвердження дієздатності мобільної платформи
Післяумови	Клієнт та сервер продовжують підтверджувати наявність зв'язку. Сервер створює план руху та починає надсилати команди руху до мікроконтролера. Команди продовжують посилатися доки відхилення від заданої точки не є меншим задовільного, або не стається втрата зв'язку.

Продовження таблиці 4.1

Опис	Після початку сесії керування та підтвердження зв'язку з сервером, користувач може подати команду на початок руху в певну точку. Користувач натискає на кнопку вибору точки руху. Кнопка візуально підкреслюється для підтвердження натискання. Після цього користувач натискає на певну точку на карті для позначення точки руху. У відповідній точці можна побачити індикатор для позначення координат. Користувач може натиснути на певну точку знову щоб змінити координати. Після того як користувач точно вирішив рухатися у певну точку, він може натиснути кнопку підтвердження руху. Індикатор на карті підкреслюється для підтвердження початку руху. Також користувач може побачити текстове повідомлення про початок руху.
Актори	Користувач системи
Тригери	Користувач натискає кнопку руху у певну точку.

Продовження таблиці 4.1

Основний потік розвитку	<ul style="list-style-type: none"> – Користувач натискає на кнопку вибору координат. – Користувач обирає координати руху. – Користувач підтверджує початок руху. – Клієнт відсилає запит на сервер. – Сервер визначає отриману команду. – Сервер розраховує план руху. – Сервер посилає команду до програми керування двигунами. – Програма керування двигунами вираховує швидкість руху, і відхилення від заданої швидкості. – Програма керування двигунами вмикає двигун відповідно до заданої швидкості. – Сервер опрацьовує візуальні дані, та використовує їх для вирахування положення рухомої платформи. – Положення у даний момент, а також швидкість двигунів використовуються як зворотній зв'язок для коректування плану руху. – Рухома платформа наближається до заданої точки. – Сервер подає команду зупинки двигунів. – Сервер очікує отримання наступної команди. – Користувач бачить повідомлення про завершення команди.
Можливі помилки	<ul style="list-style-type: none"> – Відсутність зв'язку. Користувач буде повідомлений про втрату зв'язку, клієнт спробує відновити зв'язок. Сервер, у разі неможливості відновлення зв'язку, починає повернення у початкову точку. – Відсутність зв'язку з програмою керування двигунами. Сервер перестає посылати команди, відсилає повідомлення користувачу і намагається перезапустити програму керування двигунами.

Продовження таблиці 4.1

Можливі помилки	– Втрата зв'язку з відеокамерою. Сервер перестає посилати команди, відсилає повідомлення користувачу і намагається перезапустити драйвер керування відеокамерою. У разі неможливості відновлення відеокамери, користувач отримує повідомлення і можливість почати повернення у початкову точку використовуючи лише швидкість двигунів для зворотнього зв'язку.
-----------------	--

Другою функціональною категорією є перегляд зображень з відеокамери які були створені у ході поточної сесії. Розглянемо діяльність «View camera shots». Її детальний опис наведено у таблиці 4.2.

Таблиця 4.2 – Опис діяльності «View camera shots»

Найменування	View camera shots
ID	C_02
Опис	Після початку сесії керування та підтвердження зв'язку з сервером, користувач може обрати певну точку на траєкторії та переглянути зображення, зроблене у цій точці. Користувач натискає на кнопку перегляду зроблених зображень. Траєкторія руху підкреслюється. Після цього користувач натискає на певну точку на траєкторії. У відповідній точці можна побачити індикатор для позначення координат. Користувач може натиснути на певну точку знову щоб змінити координати. При кожному натисканні користувач бачить зображення, зроблене у цій точці. Користувач може нажати на кнопку завершення перегляду для того щоб повернутися до керування.

Продовження таблиці 4.2

Актори	Користувач системи
Тригери	Користувач натискає кнопку перегляду зроблених зображень.
Передумови	Користувач розпочав сесію керування та отримав підтвердження дієздатності мобільної платформи
Післяумови	Клієнт та сервер продовжують підтверджувати наявність зв'язку. Сервер дістає дані поточної сесії, а також шлях до конкретного зображення з бази даних. Сервер відкриває необхідне зображення та надсилає його до користувача. Перегляд продовжується поки не надходить підтвердження від користувача про завершення перегляду.
Основний потік розвитку	<ul style="list-style-type: none"> – Користувач натискає на кнопку перегляду зображень. – Користувач обирає координати зображення. – Клієнт відсилає запит на сервер. – Сервер надсилає запит на отримання шляху до зображення з бази даних. – Сервер завантажує зображення в операційну пам'ять. – Сервер надсилає зображення і додаткову інформацію до клієнта. – Клієнт виводить на екран зображення а також час та місце де воно було зроблене. – Користувач натискає на кнопку завершення перегляду. – Клієнт повертається у режим керування.

Продовження таблиці 4.2

Можливі помилки	<p>– Відсутність зв'язку. Користувач буде повідомлений про втрату зв'язку, клієнт спробує відновити зв'язок. Сервер, у разі неможливості відновлення зв'язку, починає повернення у початкову точку.</p> <p>– Втрата даних. Користувач отримує текстове повідомлення про неможливість перегляду даного зображення.</p>
-----------------	---

Третьою функціональною категорією є повернення у початкову точку. Розглянемо діяльність «Return to starting point». Її детальний опис наведено у таблиці 4.3.

Таблиця 4.3 – Опис діяльності «Return to starting point»

Найменування	Return to starting point
ID	C_03
Актори	Користувач системи, сервер.
Тригери	Користувач натискає кнопку повернення у початкову точку, повна втрата зв'язку з клієнтом
Опис	Після початку сесії керування та підтвердження зв'язку з сервером, користувач може подати команду на повернення у початкову точку. Користувач натискає на кнопку повернення у початкову точку. Рухома платформа починає рухатися у початкову точку використовуючі вбудовані системи орієнтації у просторі. Також користувач може побачити текстове повідомлення про початок руху.

Продовження таблиці 4.3

Основний потік розвитку	<ul style="list-style-type: none"> – Користувач натискає на кнопку перегляду зображень. – Користувач обирає координати зображення. – Клієнт відсилає запит на сервер. – Сервер визначає отриману команду. – Сервер розраховує план руху. – Сервер посилає команду до програми керування двигунами. – Програма керування двигунами вираховує швидкість руху, і відхилення від заданої швидкості. – Програма керування двигунами вмикає двигун відповідно до заданої швидкості. – Сервер опрацьовує візуальні дані, та використовує їх для вирахування положення рухомої платформи. – Положення у даний момент, а також швидкість двигунів використовуються як зворотній зв'язок для коректування плану руху. – Рухома платформа наближається до заданої точки. – Сервер подає команду зупинки двигунів. – Сервер очікує отримання наступної команди. – Користувач бачить повідомлення про завершення команди.
Передумови	Користувач розпочав сесію керування та отримав підтвердження дієздатності мобільної платформи
Післяумови	Клієнт та сервер продовжують підтверджувати наявність зв'язку. Сервер створює план руху та починає надсилати команди руху до мікроконтролера. Команди продовжують посилатися доки відхилення від заданої точки не є меншим задовільного.

Продовження таблиці 4.3

Можливі помилки	<p>– Відсутність зв'язку з програмою керування двигунами. Сервер перестає посилати команди, відсилає повідомлення користувачу і намагається перезапустити програму керування двигунами.</p> <p>– Втрата зв'язку з відеокамерою. Сервер перестає посилати команди, відсилає повідомлення користувачу і намагається перезапустити керування відеокамерою. У разі неможливості відновлення відеокамери, сервер спробує повернутися у початкову точку використовуючи лише швидкість двигунів для зворотнього зв'язку.</p>
-----------------	---

Четвертою функціональною категорією є зміна координат початкової точки. Розглянемо діяльність «Set the return point». Її детальний опис наведено у таблиці 4.4.

Таблиця 4.4 – Опис діяльності «Set the return point»

Найменування	Set the return point
ID	C_04
Актори	Користувач системи.
Тригери	Користувач натискає кнопку встановлення початкової точки
Передумови	Користувач розпочав сесію керування та отримав підтвердження дієздатності мобільної платформи
Післяумови	Клієнт відправляє координати нової точки на сервер. Сервер зберігає нові координати точки повернення для даної сесії у базі даних.

Продовження таблиці 4.4

Опис	Після початку сесії керування та підтвердження зв'язку з сервером, користувач може подати команду на зміну координат початкової точки. Користувач натискає на кнопку зміни координат початкової точки. Після цього користувач натискає на певну точку на карті для позначення координат повернення. У відповідній точці можна побачити індикатор для позначення координат. Користувач може натиснути на певну точку знову щоб змінити координати. Після того як користувач завершив обирати координати, він може натиснути кнопку підтвердження руху.
Основний потік розвитку	<ul style="list-style-type: none"> – Користувач натискає кнопку встановлення початкової точки – Користувач обирає координати зображення. – Клієнт відсилає запит на сервер. – Сервер визначає отриману команду. – Сервер зберігає нові координати у базі даних. – Сервер відбравляє повідомлення щодо успішного збереження.
Можливі помилки	<ul style="list-style-type: none"> – Відсутність зв'язку. Користувач буде повідомлений про втрату зв'язку, клієнт спробує відновити зв'язок. Клієнт, у разі неможливості відновлення зв'язку, повідомляє користувачу про втрату зв'язку.

П'ятою функціональною категорією є початок прямого управління. Розглянемо діяльність «Begin direct control». Її детальний опис наведено у таблиці 4.5.

Таблиця 4.5 – Опис діяльності «Begin direct control»

Найменування	Begin direct control
ID	C_05
Опис	Після початку сесії керування та підтвердження зв'язку з сервером, користувач може подати команду на початок прямого управління. Користувач натискає на кнопку початку прямого управління. Після цього клієнт відкриває інтерфейс прямого керування. Карта руху набуває малих розмірів, і користувач може побачити зображення з відеокамери у прямому часі. Для прямого управління користувачу доступно чотири кнопки – вперед, назад, вліво та вправо. Також є кнопка завершення прямого керування, при натисненні якої інтерфейс повертається у початкове положення.
Актори	Користувач системи.
Тригери	Користувач натискає кнопку початку прямого управління.
Основний потік розвитку	<ul style="list-style-type: none"> – Користувач натискає кнопку початку прямого управління. – Клієнт відсилає запит на сервер. – Сервер розпочинає пряму трансляцію зображень з відеокамери. – Сервер відправляє повідомлення про початок керування. – Користувач бачить зміну інтерфейсу у режим прямого управління. – Клієнт продовжує постійно відправляти запити для підтвердження зв'язку. – Користувач натискає кнопку завершення прямого керування.

Продовження таблиці 4.5

Передумови	Користувач розпочав сесію керування та отримав підтвердження дієздатності мобільної платформи
Післяумови	Клієнт натискає кнопку завершення прямого керування.
Основний потік розвитку	<ul style="list-style-type: none"> – Користувач натискає кнопку початку прямого управління. – Клієнт відсилає запит на сервер. – Сервер розпочинає пряму трансляцію зображень з відеокамери. – Сервер відправляє повідомлення про початок керування. – Користувач бачить зміну інтерфейсу у режим прямого управління. – Клієнт продовжує постійно відправляти запити для підтвердження зв'язку. – Користувач натискає кнопку завершення прямого керування.
Можливі помилки	<ul style="list-style-type: none"> – Відсутність зв'язку. Користувач буде повідомлений про втрату зв'язку, клієнт спробує відновити зв'язок. Сервер посилає команду зупинки двигунів, та очікує на відновлення зв'язку. Якщо відновити зв'язок немає можливості, сервер розпочинає рух до точки повернення.

Інші активності мають схожі властивості та можливі випадки поведінки, тому їх детальний огляд буде опущено.

4.2 Система передачі команд управління рухомою платформою

Як уже було розглянуто, однією з основних небезпек використання мобільних платформ є втрата зв'язку. Важливим завданням у такому випадку є забезпечення зупинки. У даній роботі була використана математична модель кінцевих автоматів на всіх рівнях архітектури – клієнта, сервера та мікроконтролера. У випадку втрати зв'язку кінцевий автомат повертає свій стан у стан спокою, зупиняючи платформу.

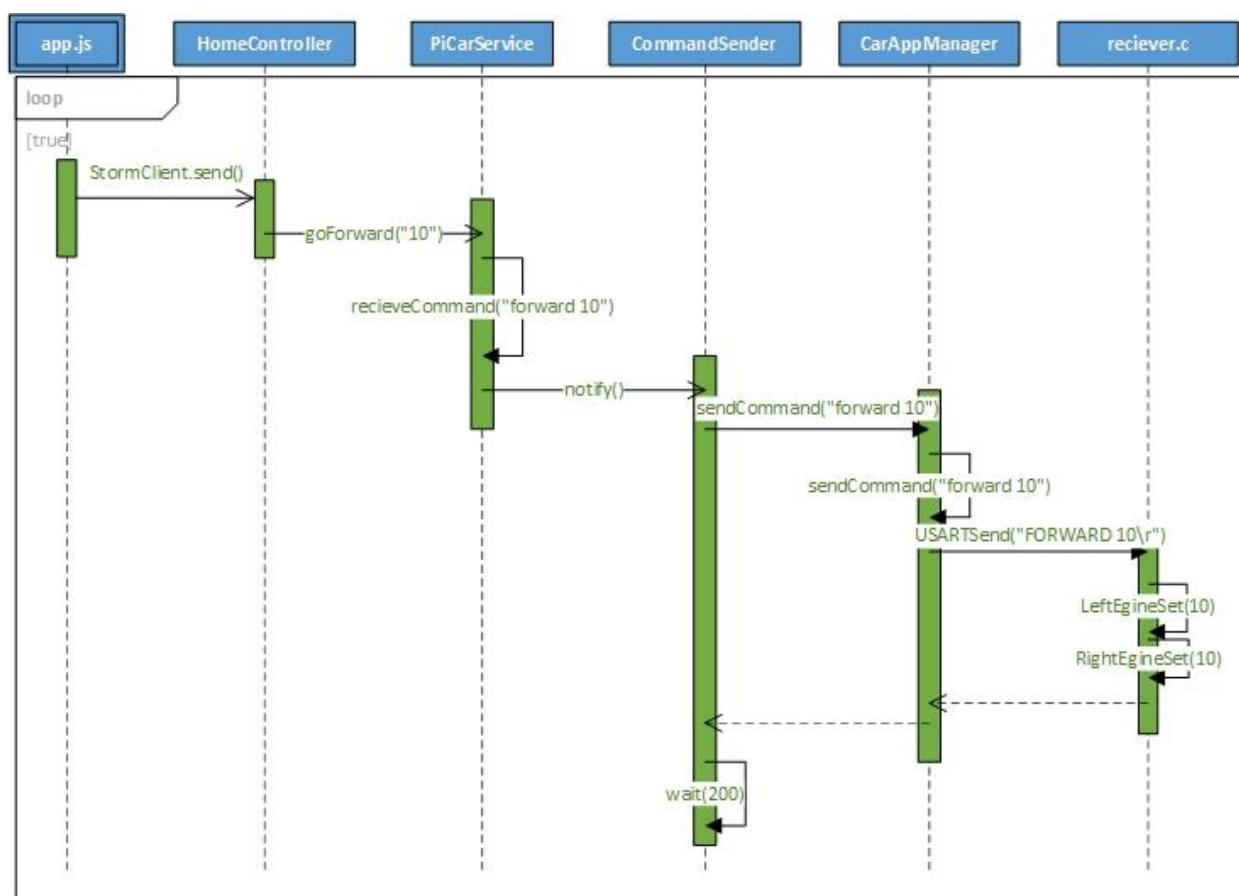


Рисунок 4.2 – Діаграма послідовності передачі команд керування.

Клієнт(рисунок 4.2) працює в циклі під час прямого керування мобільною платформою. У цьому циклі клієнт посилає стан свого кінцевого автомата на сервер і чекає на підтвердження виконання команди від сервера. Логіка виконання команди поділена на класи `CommandController`, `PiCarService`, `CommandSender` та `CarAppManager`, відповідно до архітектури MVC що використовується на сервері. Розглянемо роботу системи прямого керування

більш детально за допомогою UML діаграми послідовності передачі команд керування, де для прикладу взято команду руху вперед.

Інкапсуляція обробки запиту виконується у класі `CommandController`, що викликає клас `ContinuousCarService` для подальшої його обробки. Після того як клієнт робить запит до серверу, клас контролера відповідає за повернення відповіді на цей запит, виконуючи зазвичай одну дію. Таким чином, у ході виконання дії контролер може повертати різні типи результатів. Контролер отримує дані від користувача за допомогою клієнта. Потім сервіс обробляє дані користувача за допомогою моделі, і контролер повертає результат. Заради забезпечення надійності роботи програми, у разі отримання неправильної команди сервіс посилає команду зупинки до мікроконтролера.

Клас `ContinuousCarService` відповідає за обробку конкретних команд. Математичною основою для даного класу є кінцевий автомат. Цей клас відповідає за трансформацію клієнтських команд у команди для мікроконтролера. Забезпечення перевірки наявності зв'язку відбувається за допомогою регулярних команд від клієнта, що надсилають теперішній стан кінцевого автомата, які надсилаються кожні 100 мс. Через те що відмовостійкість зв'язку між сервером та мікроконтролером перевищує відмовостійкість зв'язку між клієнтом та сервером на багато порядків, надсилання регулярних команд до мікроконтролера не є необхідним. Тому команди до програми керування двигунами(далі ПКД), яка знаходиться на мікроконтролері, поступають тільки в разі зміни стану кінцевого автомату сервера.

Задля впровадження усіх поставлених варіантів використання обробка стану серверного автомата має бути незалежною від іншої логіки сервера, тому вона має обчислюватись в окремому потоку. Для виконання даної вимоги використовується допоміжний клас `CommandSender`. Багатопотоковість досягається створенням об'єкту класу `CommandSender` за допомогою класу `ContinuousCarService`. `ContinuousCarService` надсилає усі команди до `CommandSender`. Після отримання команди клас `CommandSender` чекає

отримання наступної. Оскільки клієнт надсилає неперервний потік команд, а ПКД потребує команди лише у разі зміни стану, CommandSender перевіряє нову команду з попередньою, і у разі зміни стану надсилає цю команду до ПКД. Якщо нова команда не поступила від сервера протягом 200 мс, вважається що зв'язок не є стабільним для забезпечення керування або втрачений, і ПКД отримує команду зупинки.

Програма керування двигунами написана мовою програмування C, і виконується на мікроконтролері STM32F103C8T6. Обмін інформацією виконується за допомогою протоколу USART, що дозволяє серверу та ПКД обмінюватися текстовими повідомленнями у реальному часі. Архітектурно ПКД є кінцевим автоматом, що змінює свій стан залежно від надісланих команд. Клас CarAppManager інкапсулює роботу з ПКД, надаючи інтерфейс надсилання команд, що використовується ContinuousCarService.

Розглянемо роботу ПКД більш детально за допомогою UML діаграми станів.

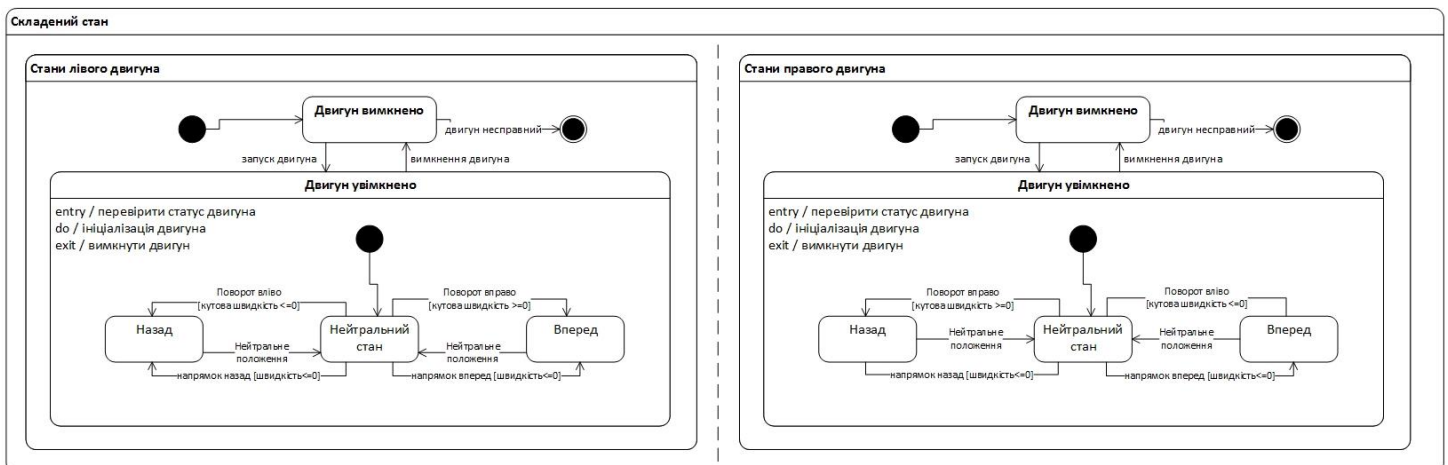


Рисунок 4.3 – Діаграма станів програми керування двигунами

Як ми можемо побачити на рисунку 4.3, на відмінну від сервера, у якому стани кінцевого автомата є простим переліком – «зупинка», «рух уперед», «поворот наліво», тощо – логіка ПКД реалізована за допомогою ієрархії кінцевих автоматів, які реалізують логіку роботи кожного двигуна окремо.

4.3 Система передачі поточного відео

Розглянемо роботу системи за допомогою UML діаграми послідовності передачі поточного відео.

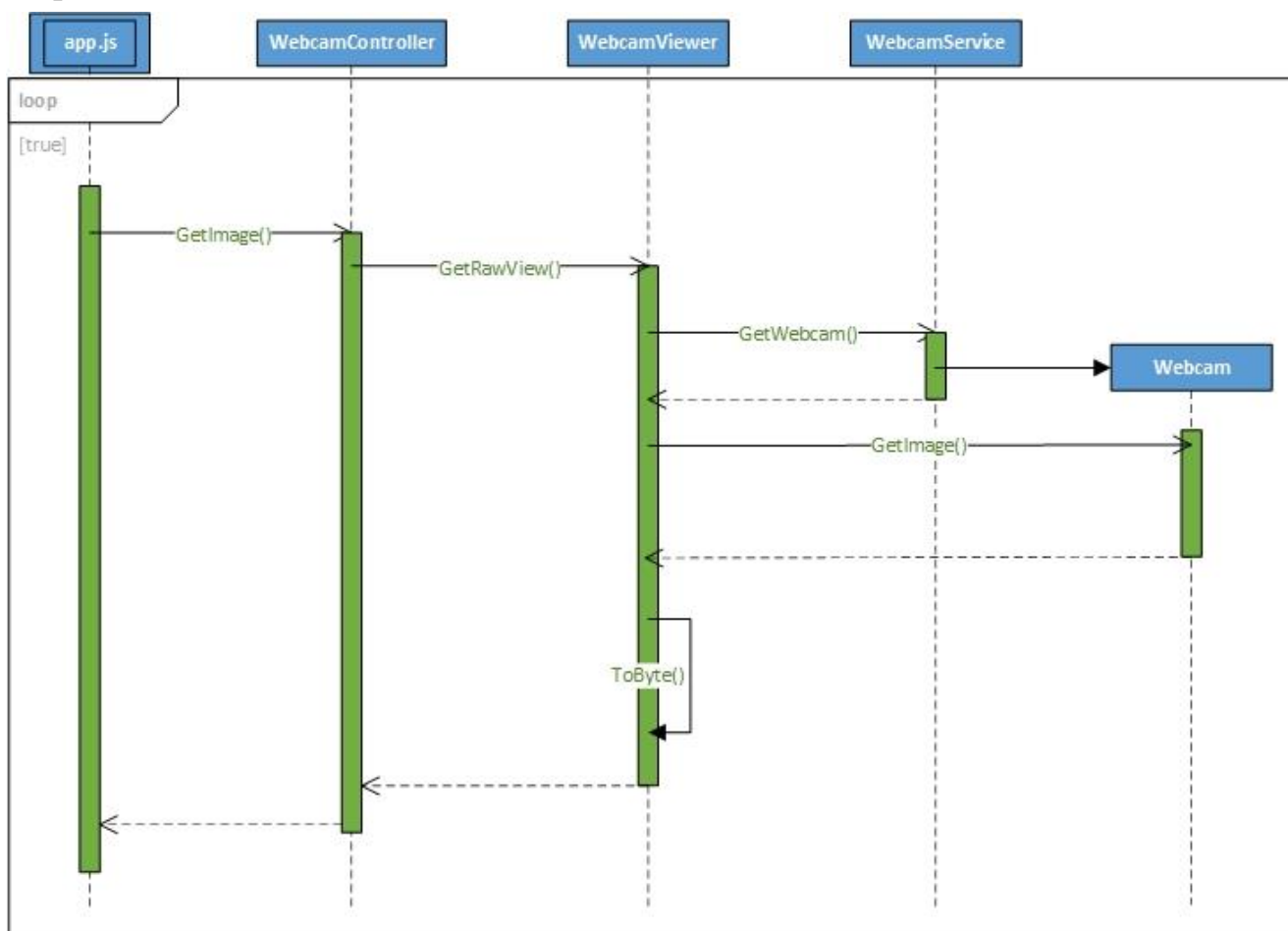


Рисунок 4.3 – Діаграма послідовності передачі поточного відео.

Під час виконання сценарію прямого управління мобільною платформою, клієнт app.js(рис 4.3) надсилає неперервний потік зображень з відеокамери у циклі. Завдяки тому, що цикл виконується тридцять разів на секунду, клієнт може показувати відео у реальному часі.

Робота циклу починається з того що клієнт посилає запит на отримання зображення на сервер. Відповідно до архітектури MVC бізнес-логіка отримання зображення поділена на класи WebcamController, WebcamViewer та WebcamService. Розглянемо їх роботу детальніше.

Інкапсуляція обробки запиту виконується у класі WebcamController, що викликає клас WebcamViewer для подальшої його обробки. Після того як клієнт

робить запит до серверу, клас контролера відповідає за повернення відповіді на цей запит, виконуючи зазвичай одну дію. Таким чином, у ході виконання дії контролер може повертати різні типи результатів. Контролер отримує дані від користувача за допомогою клієнта. Потім сервіс обробляє дані користувача за допомогою моделі, і контролер повертає результат.

Клас `WebcamService` відповідає за налаштування роботи камери. Для цього клас `WebcamService` зчитує налаштування сервера і створює `Webcam` відповідно до зазначених специфікацій, а саме розширення відео та драйвер, використаний для роботи з камерою.

Клас `WebcamViewer` відповідає за обробку запиту, інкапсулюючи процес отримання зображення від сервера. До створеного до цього об'єкту класу `Webcam` виконується запит отримання зображення. За відсутності зображення `WebcamViewer` повертає пустий квадрат, а `WebcamService` додає до нього повідомлення до клієнта про відсутність зв'язку з відеокамерою. У разі успішного отримання зображення воно конвертується у масив байтів за допомогою стандартного класу `ByteArrayOutputStream`.

4.4 Інтегрована система збереження даних системи управління рухомої платформи

Для СУРП однією з основних вимог є наявність інтегрованої системи збереження отриманої інформації. Вона є необхідною в тому випадку, коли внаслідок відсутності зв'язку СУРП починає працювати автономно.

Через те, що система збереження інформації є актуальною для мобільних платформ, була розроблена і обґрунтована архітектура інтегрованої підсистеми збереження даних СУРП. Розглянемо роботу системи за допомогою ER діаграми, зображеної на рисунку 4.4.

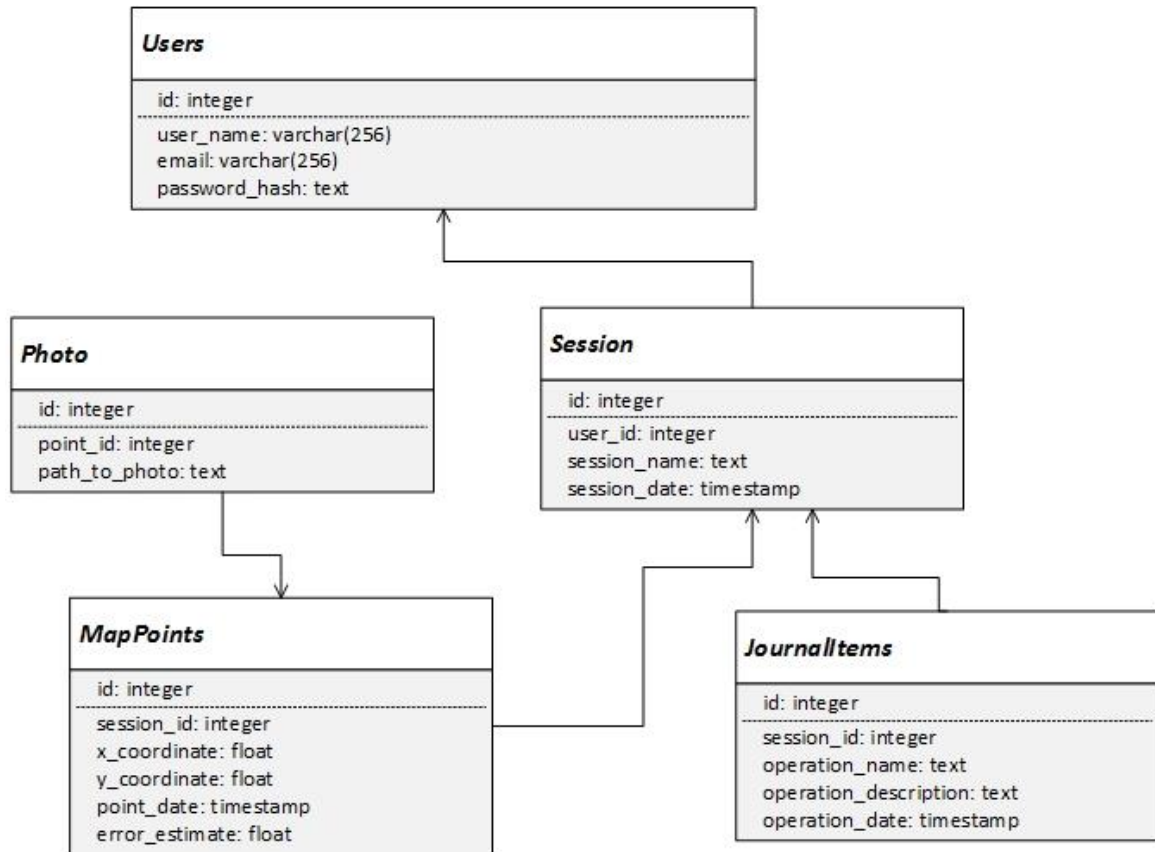


Рисунок 4.3 – ER діаграма бази даних

Система зберігає п'ять сутностей даних – користувачі, сесії, записи журналу, точки карти та зображення.

Усі отримані дані групуються в сесії. Сесія являє собою єдину сутність, що об'єднує у собі усі дані, отримані в єдиний період часу, наприклад, під час дослідження однієї кімнати. Для підвищення швидкодії сесія зберігається в оперативній пам'яті. Після закінчення дослідження території сесія завантажується у інтегровану базу даних. Клієнт у будь-який час може вивести список усіх сесій, і обрати необхідну йому сесію для аналізу процесу дослідження певної зони.

Усі сесії групуються відповідно до користувачів які виконували дослідження. Відповідно, сутність користувача зберігає інформацію, необхідну для авторизації.

У процесі дослідження території усі важливі моменти дослідження, такі як надіслані команди, відмови в роботі системи, тощо зберігаються як записи у журнал сесії, і можуть бути переглянути після завершення роботи.

Під час роботи відеокамери алгоритм локалізації на основі відеоданих знаходить положення СУРП у певний момент, разом з очікуваною помилкою. Ці дані зберігаються для того щоб користувач міг відтворити пройдений шлях в процесі виконання певної сесії при подальшому аналізі.

Певні опорні точки також зберігають зображення, отримане у певний момент часу. Таким чином, користувач може побачити пройдений шлях, та зображення, які були отримані в процесі роботи СУРП.

4.5 Моделювання відмовостійкості за допомогою марковських процесів

Для визначення інтенсивності відмов використаємо дані дослідження відмов мобільних платформ[21]. Види відмов характеризуються на рисунку 4.4.

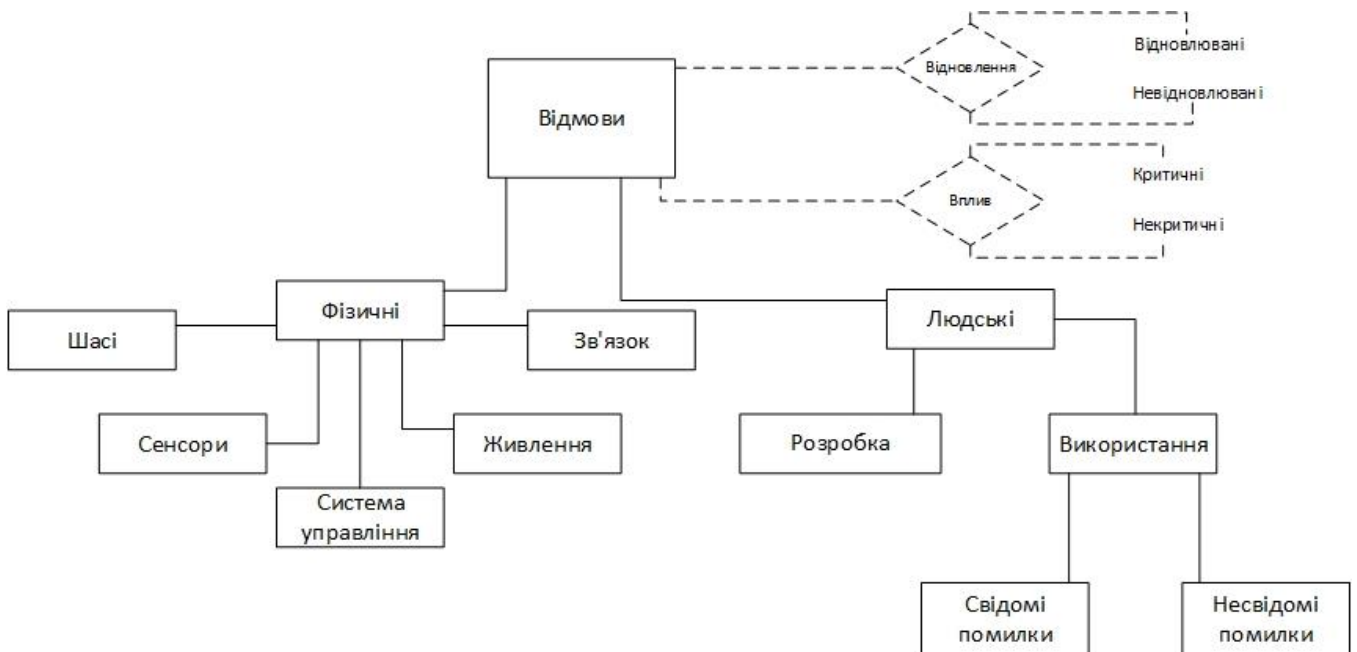


Рисунок 4.4 – Види відмов автоматизованих систем управління рухомими платформами.

Використаємо дані дослідження повторного дослідження інтенсивності фізичних відмов автоматизованих систем управління рухомими платформами, проведене у роботі[21], оскільки вони надають найновіші і найточніші дані.

Таблиця 4.5 – Вірогідності фізичних відмов за класом[21]

Виробник	Шасі	Система управління	Живлення	Зв'язок	Сенсори
Inuktun	0.45	0.32	0.00	0.02	0.12
iRobot	0.22	0.30	0.15	0.22	0.11
Всього	0.36	0.31	0.06	0.10	0.12

Із таблиці 4.5 ми можемо зробити висновок що відмови зв'язку в середньому для СУРП складають 10% від усіх видів відмов.

Припустимо, що частоти відмов усіх підсистем є постійними. У такому випадку, знаючи середній час до відмови всієї системи, ми можемо вирахувати середню частоту відмови систем зв'язку СУРП.

Таблиця 4.6 – Частота відмов і час до відмови[21]

Виробник	Відмов	Відмов за годину	Час до відмови(годин)
Inuktun	58	0.12	8.74
iRobot	89	0.06	15.77
Всього	147	0.09	13.23

Відповідно до таблиці 4.6, ми можемо вирахувати загальну інтенсивність відмов СУРП на теперішній час, що складає:

$$\lambda_z = \frac{1}{13.23} = 0,076 \text{ (1/год)}$$

Знаючи що відмови зв'язку складають 0.1 від усіх відмов СУРП, ми можемо вирахувати загальну інтенсивність відмов зв'язку СУРП:

$$\lambda = 0,076 * 0.1 = 0,0076 \text{ (1/год)}$$

Припущення і обмеження. У більшості випадків резервування систем зв'язку відбувається в режимі гарячого резерву. Припустимо, що технічні і надійнісні характеристики радіоканалу та каналу Wi-Fi мають однакові чисельні значення. Під час відмови одного з каналів автоматично підключається резервний.

Часом переключення будемо зневажати, оскільки час підключення значно менший у порівнянні з часом роботи системи.

Надійність інших елементів системи є меншою за систему зв'язку, проте їх резервування не є практичним через їх габарити та ціну. Тому їх резервування не буде розглядатися.

Постановка задачі.

Як було визначено, інтенсивність відмов λ для радіоканалу має середнє значення 0,0076 (1/год) у польових умовах. Зважаючи з приведених припущень дана система представляє собою 2 паралельно працюючих канали зв'язку. Один з даних каналів є основним інший у гарячому резерві. Таким чином система може перебувати у трьох станах:

1. S2 – система справна, має 1 канал у гарячому резерві та 1 основний;
2. S1 – один резервний канал не справний, 1 основний;
3. S0 – система повністю несправна і відновлюється.

З урахуванням припущень отримаємо надійнісні характеристики технічної системи.

Постановка задачі передбачає що для її рішення можна використовувати схему чистої загибелі.

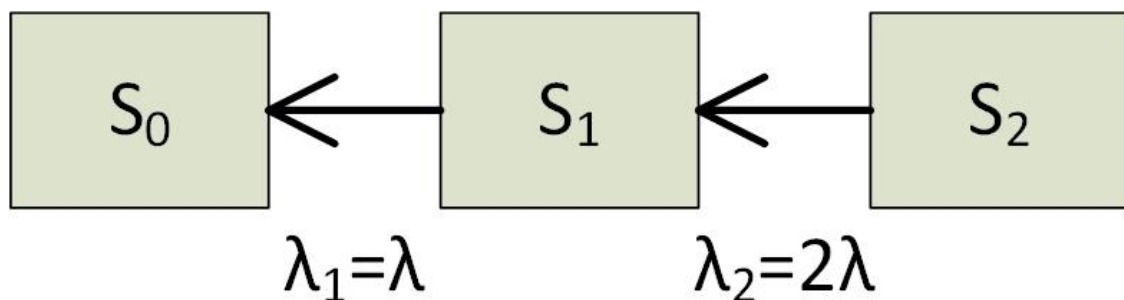


Рисунок 4.5 – Граф станів процесу загибелі для 3 каналів зв'язку

Візьмемо частоту відмови λ :

$$\lambda = 0,0076 \frac{1}{\text{год}}$$

Тоді вірогідність безвідмовної роботи n каналів зв'язку[22]:

$$p_n(t) = e^{-n\lambda t}$$

Для двох каналів[22]:

$$p_2(t) = e^{-2\lambda t} = e^{-2*0,0076*t}$$

Вірогідність відмови n каналів[22]:

$$p_{n-i}(t) = \frac{n!}{(n-i)!} \left(\sum_{k=0}^i e^{-\lambda(n-k)t} / \left[\sum_{l=0}^i \prod_{h=0}^i (k-h)/(k-l) \right] \right)$$

Відповідно, вірогідність відмови одного з каналів[22]:

$$p_2(t) = 2 * [e^{-1\mu t} - e^{-2\mu t}]$$

У такому випадку, вірогідність повної відмови[22]:

$$p_0(t) = 1 - \sum_{i=1}^n i p_{n-i}(t)$$

Побудуємо графіки розподілу імовірності відмови з часом.

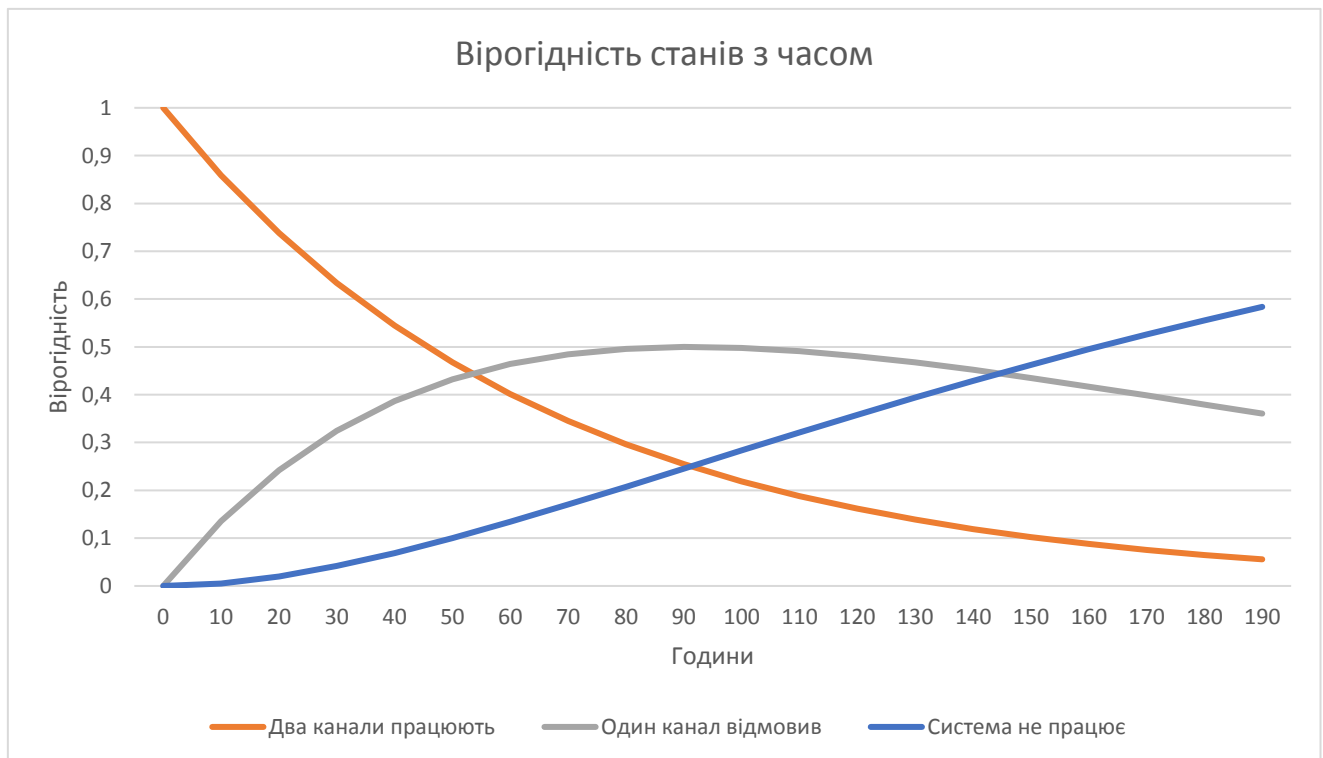


Рисунок 4.6 – Вірогідність станів системи зв'язку з часом

Як можна побачити на рисунку 4.6, резервування системи зв'язку зменшує вірогідність одночасної роботи усіх систем. Проте, у той же самий час, вірогідність повної відмови зростає значно повільніше.

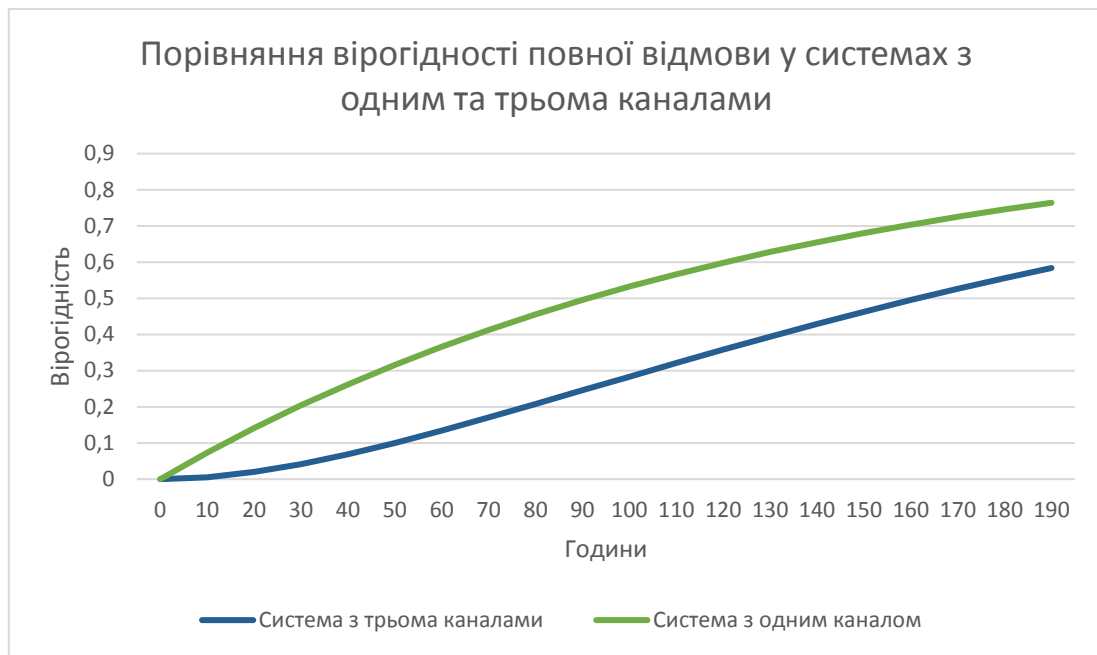


Рисунок 4.7 – Порівняння вірогідності повної відмови у систем з одним та трьома каналами.

Ми можемо побачити на рисунку 4.7, де чітко видно, що у порівнянні з системою з одним каналом, вірогідність відмови є значно нижчою протягом усього періоду використання. Додатково ми можемо побачити що на початку вірогідність повної відмови зростає значно повільніше. Розглянемо це співвідношення докладніше.

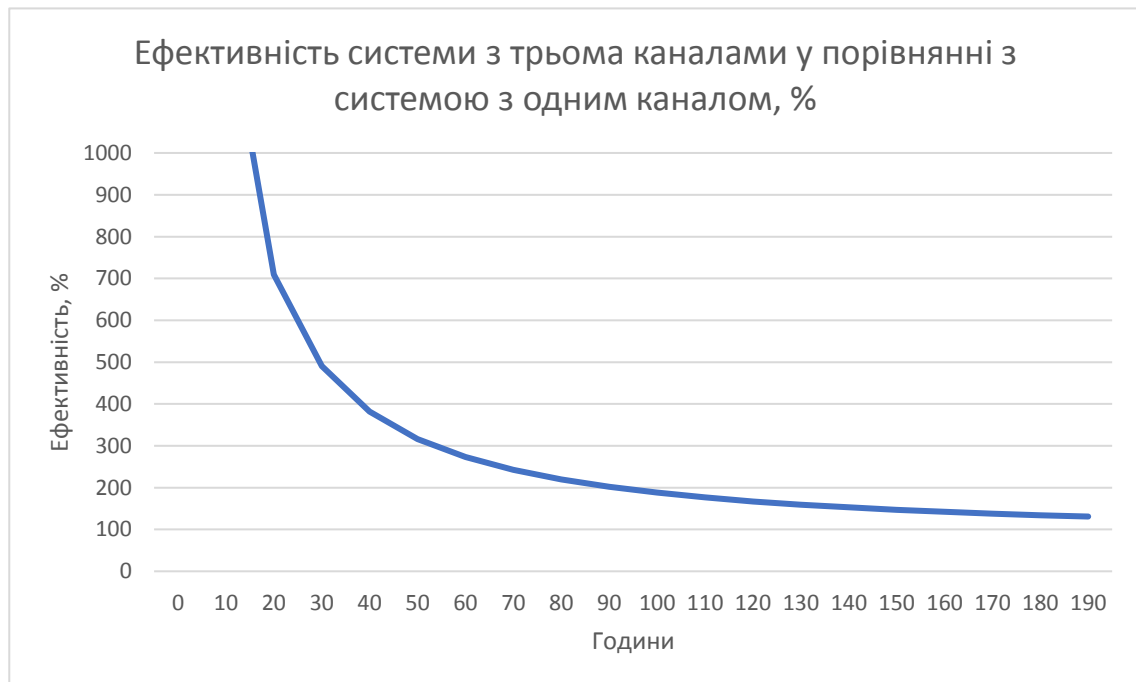


Рисунок 4.8 – Ефективність системи з трьома каналами у порівнянні з системою з одним каналом.

Як ми можемо побачити на рисунку 4.8, вірогідності повної відмови є значно меншими на початку використання. Із цього можемо зробити висновок, що використання резервування системи зв'язку є більш ефективним у випадку, коли система зв'язку використовується менше, ніж її середній час до відмови. Враховуючи, що система зв'язку не є основною причиною відмов СУРП, і те що середній час до відмови СУРП є значно меншим ніж середній час до відмови системи зв'язку, можна зробити висновок, що резервування системи зв'язку дозволить майже повністю прибрати ймовірність відмови системи зв'язку протягом середнього часу використання СУРП. Таким чином, враховуючи той факт що система зв'язку становить незначну частину вартості рухомої платформи, можна сказати що її резервування є доцільним.

4.6 Висновки

Була розроблена і описана Use Case діаграма рухомої платформи. Описані типові сценарії використання системи, такі як Set coordinates, View camera shots, Return to starting point, Set the return point, та Begin direct control. Для систем передачі команд управління та передачі потокового відео були розроблені UML діаграми послідовності, та описано принципи їх роботи. Була описана інтегрована система збереження даних. Побудовано UML діаграму сутностей для опису інтегрованої системи збереження даних. Було проведено моделювання резервування системи зв'язку з використанням математичного апарату марковських процесів. За допомогою математичної моделі була доведена доцільність резервування системи зв'язку.

5 СТАРТАП ПРОЕКТ

В даному розділі проводиться опис рішення для впровадження та монетизації прототипу, розробленого в даній магістерській дисертації. Буде проведено аналіз рішень що вже існують на ринку. На основі цього аналізу будуть сформовані вимоги та ідеї для впровадження розробленого прототипу. Необхідно також розглянути переваги та недоліки прототипу для його комерціалізації.

5.1 Опис ідеї проекту

В ході розробки стартап проекту була поставлена мета – автоматизована рухома платформа на основі фреймворку Spring, що забезпечує надійний, безперервний зв'язок для керування рухомою платформою, мінімізує потребу у виділенні людських ресурсів для нагляду за системою, що значно зменшує вартість утримання системи і підтримує роботу системи в автономному режимі. Зміст ідеї, напрями застосування, а також їх переваги наведено у таблиці 5.1.

Таблиця 5.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
	1. Збір та аналіз даних про середовище небезпечної зони	Автоматизування процесу збору даних
	2. Забезпечення часткової роботи системи в автономному режимі	Зменшення навантаження на людські ресурси

Продовження таблиці 5.1

Зміст ідеї	Напрямки застосування	Вигоди для користувача
	3. Резервування системи зв'язку.	Зменшення часу на відновлення СУРП, зменшення витрат на втрачені платформи
	4. Вбудована база даних	Спрощення фази аналізу та обчислення результатів дослідження.

Під час проведення техніко–економічного аналізу проведено порівняння ряду характеристик стартап проекту з характеристиками систем конкурентів та аналогів. За критерії порівняння взято техніко–економічні характеристики ідеї, а саме визначення їх переліку для даного стартап проекту та для проектів–конкурентів.

Аналіз ринку показує, що у даної системи немає потенційних конкурентів. Аналогом може слугувати поєднання схожих рішень, наприклад, JetBot та Freenove 4WD Smart Car. Проте впровадження даного варіанту є досить складним, документація відсутня, а поріг для розуміння зазначених рішень є досить високим. Більш докладно описані слабкі, нейтральні та сильні сторони проектів у таблиці 5.2.

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

Техніко– економічні характерист ики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
	Дана система	JetBot	Freenove 4WD Smart Car agalix			
Ліцензія	Безкош- товна	Безкош- товна	Платна			+
Мова	Java	Python	Python		+	
Графічний інтерфейс	Є	Відсутній	Є			+
Робота в автомати- зованому режимі	Є	Є	Відсутня			+
Моніторинг подій	Є	Частково	Відсутня			+
Підтримка окремих подів	Відсутня	Є	Відсутня	-+		

Продовження таблиці 5.2

Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
	Дана система	JetBot	Freenove 4WD Smart Car agalix			
Патерни навантаження	Є	Відсутні	Відсутні			+

На основі зведеної таблиці порівняння техніко-економічних характеристик можна визначити слабкі, нейтральні та сильні сторони сформованої ідеї проекту стартапу. Аналізуючи співвідношення сильних сторін даного потенційного продукту з властивостями систем-конкурентів, можна зробити висновок про конкурентоспроможність даного продукту

5.2 Технологічний аудит ідеї проекту

Наступним етапом розробки стартап проекту є аудит технологій для визначення реалізації складових компонентів. Серед використаних технологій – мова програмування Java, для розгортання використовується фреймворк Spring, для роботи з візуальною інформацією використовується OrbSlam2.

Дана технологічна сфера дуже швидко розвивається – одні проекти помирають і на зміну приходять нові. Саме тому дуже важливо обрати актуальні технології, для того, щоб проект не став застрілим з початку розробки. Обрані технології, а саме мова програмування Java і фреймворк Spring є новітніми та актуальними засобами.

В таблиці 5.3 проаналізовано всі використані при розробці компоненти для описання технологічної втілюваності ідеї:

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технології	Доступність технологій
1		Мова програмування Java	Є в наявності	Доступні безкоштовно
2		Бібліотека для візуальної одометрії OrbSlam2	Є в наявності	Доступні безкоштовно
3		Фреймворк Spring	Є в наявності	Доступні безкоштовно
Обрані технології реалізації проекту включають в себе мову програмування Java, яка виконує візуальну одометрію інструментами бібліотеки OrbSlam2. Як основний ферймворк обрано Spring. Всі інструменти наразі безкоштовні.				

5.3 Аналіз ринкових можливостей запуску

Дуже важливим підготовчим етапом перед впровадженням стартап проекту є аналіз ринкових можливостей та ринкових загроз. Виходячи зі стану розвитку ринку у даній галузі, можна сформулювати напрямок розвитку, задовольняючи потреби потенційних клієнтів. Серед перешкод у реалізації проекту варто враховувати потенційні ринкові загрози і конкурентоспроможність даного проекту відносно набору пропозицій конкурентів(таблиця 5.4).

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	12
2.	Загальний обсяг продаж, грн/ум.од	5 ум.од.
3.	Динаміка ринку (якісна оцінка)	Зростає
4.	Наявність обмежень для входу (вказати характер обмежень)	Початковий капітал до 30 тис. ум. од
5.	Специфічні вимоги до стандартизації та сертифікації	Формально відсутні
6.	Середня норма рентабельності в галузі (або по ринку), %	78

Наступним кроком аналізу ринку є визначення актуальних потреб користувачів та відношення їх до груп цільової аудиторії. В залежності від потенційної групи цільових клієнтів, вимоги до споживчих товарів можуть змінюватись. Зведена таблиця характеристик потенційних клієнтів наведена в таблиці 5.5.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Стрімкий розвиток розподілених систем	Компанії-постачальники програмного забезпечення	Доменна область поставки програмного забезпечення	Простота інтеграції до існуючої інфраструктури
2	Розподілення ресурсів сервісу від навантаження	Розробники програмного забезпечення.	Умови для застосування автоматизованої зміни лімітів для сервісу	Зручне відображення актуальних метрик

Визначивши потенційні групи клієнтів, проведено аналіз ринкового середовища. Таблиці 5.6 демонструє потенційні фактори загроз та розриває їх суть. У таблиці 5.7 будуть наведені фактори можливостей, які сприяють майбутньому розвитку стартап проекту як конкурентоспроможної одиниці.

Таблиця 5.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Поява більш продуктивних методів	Можлива поява та стрімкий розвиток нового методу, що матиме кращі характеристики	Спроба підвищити якість системи
2.	Відсутність інвестицій	Відсутність коштів унеможливило розвиток за рахунок відсутності ресурсів	Пошук нових джерел коштів
3.	Ціна ліцензії	Об'єм ресурсів для розробки та тестування прямо пропорційно впливають на ціну товару	Перегляд ціни за рахунок пошуку більшої кількості покупців

Фактори можливостей - сприятливі фактори зовнішнього середовища, які можуть впливати на зростання бізнесу в майбутньому. Значення можливостей ринку для компанії в стратегічному плануванні: можливості ринку уособлюють джерела зростання бізнесу. Можливості необхідно аналізувати, оцінювати і розробляти план заходів по їх використанню з залученням сильних сторін компанії. Фактори які є актуальними для даного стартап проекту – це укладення угоди з розвиненими інформаційними компаніями, отримання інвестицій на розвиток проекту, а також інтеграція з однією з розвинених компаній.

Таблиця 5.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Підписання контракту	Укладення договору про інтеграцію даного стартап проекту в існуючу систему	Збільшення ресурсів для розробки
2.	Інвестиції	Отримання коштів від інвесторів	Вкладення інвестицій у нові технології
3.	Інтеграція з ІТ компанією	Інтеграція в систему розвиненої компанії	Постійна фінансова підтримка

Як видно з проаналізованих даних, конкуренція як така відсутня, тобто є можливість вільно посісти це місце на ринку. Є також частково готові рішення: деякі необхідні бібліотеки є у вільному доступі, щоб використовувати їх для розробки конкретної ліцензованої програмної одиниці.

Основними перевагами при виборі цього товару є простота використання, широкий спектр функцій та якість роботи, якої немає у конкурентів.

Привабливість розвитку на даному ринку додає досить слабка конкуренція та неширокий спектр подібних продуктів. Також не виникає проблем з використанням частково підготовлених рішень, адже модулі СУРП надаються на основі безкоштовної ліцензії або умовно-безкоштовним модулем використання. Розвиток технологічності позиціонується як основна сильна сторона, на яку конкуренти не мають впливу.

Аналізуючи ступінь конкурентності на ринку, взято до уваги фактори маркетингового середовища, концепцію ідеї проекту, ключові потреби користувача та критерії їх вибору. Враховуючи цю інформацію, можна визначити конкурентні фактори. Фактори наведені у таблиці 5.9.

В таблиці 5.8 розглядаються фактори конкурентоспроможності, а саме наявність графічного інтерфейсу, легкість у використанні, точність звітування та орієнтовність на потреби користувача. Орієнтовність на потреби користувача вважаю

ключовою особливістю, адже конкуренти не можуть цього забезпечити. Легкість використання також є важливою рисою даного стартап-проекту.

Таблиця 5.8 – Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Зручність інтерфейсу	Проводячи аналіз конкурентних товарів, можна зробити висновки на основі відгуків користувачів та виділити недоліки, які не потрібно допускати та переваги, які можна прийняти до уваги на етапі розробки проекту.
2	Забезпечення інтуїтивної зрозумілості	Так як система передбачає взаємодію з користувачем, буде забезпечено максимально лаконічний та зрозумілий інтерфейс з навігацією розділів.

Продовження таблиці 5.8

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
3	Точність звітування	Для того щоб максимально задовольнити потребу адаптації системи до умов навантаження, точність звітування має бути високою
4	Орієнтованість на потреби користувача	Певні потреби користувачів досі залишаються не покритими, тому дана система виділяється серед представлених на ринку своє орієнтованістю на поточні потреби

Після визначення факторів конкурентоспроможності, виділено сильні та слабкі сторони стартап-проекту (СП – стартап-проект, К – конкурент). Дані наведені у таблиці 5.9.

Таблиця 5.9 – Порівняльний аналіз сильних та слабких сторін даного проекту

№ п/п	Критерій конкурентоспроможності	Оцінка 1-30	Порівняння існуючих рішень у конкурентів						
			-3	-2	-1	0	+1	+2	+3
1	Швидкість впровадження	30			+				
2	Якість результатів роботи проекту	25			+				
3	Додаткова конфігурація	27	+						
4	Звітність	25				+			

Наступним етапом аналізу ринку та можливості запуску проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities). SWOT-аналіз стартап-проекту наведено у таблиці 5.10.

Таблиця 5.10 – SWOT-аналіз стартап-проекту

<p>Сильні сторони:</p> <ul style="list-style-type: none"> - зручність інтерфейсу - економія - простота у використанні - врахування досвіду конкурентів 	<p>Слабкі сторони:</p> <ul style="list-style-type: none"> - перший вихід на ринок - наявність open source рішень - ціна за рахунок складності розробки
<p>Можливості:</p> <ul style="list-style-type: none"> - інтеграція рішення у великі існуючі системи - використання стартапами 	<p>Загрози:</p> <ul style="list-style-type: none"> - витіснення існуючими open source рішеннями - пошук клієнтів

5.4 Ринкова стратегія

Для побудови ринкової стратегії необхідно провести аналіз ринку: потреби клієнтів, потреба у даному продукті, конкурентів потенціалу ринку.

Під час аналізу ринку, насамперед, необхідно визначити товарно-географічні межі, суб'єкти ринку (продавці та клієнти), місткість ринку, а також структуру, особливо структуру суб'єктів господарювання на ринку, що характеризуються кількісними показниками, а також якісні показники структури ринку. Обов'язковим етапом проведення аналізу ринку є аналіз ринкового потенціалу ринкових суб'єктів.

Обираємо цільові групи, а саме ІТ компанії, стартап-проекти та державні ІТ організації. Всі вони зацікавлені в тому, аби зменшити економічні витрати. В перші два легкість входу найменша, а от в державному сегменті буде складніше.

Таблиця 5.11 – Цільові групи потенційних покупців

№ п/п	Цільові групи потенційних клієнтів	Зацікавленість введення проекту на ринок для споживача	Відсоток в межах цільової групи	Конкуренція в сегменті	Легкість входу у сегмент
1	ІТ компанії	Висока	70%	Середня, але згодом збільшиться	Низька складність
2	Стартап проекти	Середня	50%	Середня, але згодом збільшиться	Низька складність
3	Неприбуткові державні організації ІТ	Висока	20%	Низька	Середня складність

5.5 Висновки

Підбиваючи підсумки проведеного аналізу, можна дійти висновку, що стартап-проект на базі розробленого програмного рішення має потенційну можливість ринкового впровадження. Дана можливість підкріплюється великими шансами прибутковості, адже запроваджене рішення має достатньо переваг над представленими на ринку та може скласти їм конкуренцію. Ринок мобільних платформ для дослідження території розвивається стрімкими темпами, адже попит на робототехніку все більше зростає. Користувачі мобільних платформ шукають нові шляхи підвищення надійності та відмовостійкості у небезпечних умовах, а даний продукт покликаний саме для цього.

Були розглянуті такі показники, як сильні, слабкі та нейтральні характеристики проектів, технологічна здійсненність проекту, цільова аудиторія, попит ринку на рішення даної категорії, фінансові витрати, фактори загроз та фактори можливостей, шляхи отримання інвестицій та інше.

Подальший розвиток проекту є перспективним, враховуючи галузь впровадження, можливості, що пропонує даний проект, попит на ринку, а також відсутність монополії.

ВИСНОВКИ

У дисертації розглянуто проблему створення автоматизованої системи управління наземною рухомою платформою на основі фреймворку Spring. Проведений аналіз варіантів реалізації виділив актуальність проблеми і допоміг сформулювати вимоги і виділити основні напрямки дослідження.

Представлено комплексний підхід до вирішення проблеми, присвяченій задачі створення та автоматизації прототипу рухомої платформи для дослідження небезпечних зон. Розроблене рішення надає можливість інтеграції у все існуючу, функціонуючу систему. Перевагою рішення є впровадження сучасних систем з більшою обчислювальною потужністю, що надає можливість швидшої конфігурації а також збільшення обсягу виконуваних завдань. Створено модель надійності резервування систем зв'язку та забезпечення автономного повернення платформи у разі втрати зв'язку.

У ході виконання роботи було проведено дослідження технологій для побудови апаратної інфраструктури наземних роботів, та обґрунтовано вибір компонентів в контексті системи управління рухомою платформою. Було побудовано повнофункціональний прототип системи на основі розглянутих компонентів та доведено роботоспроможність систем передачі команд, передачі потокового відео та збереження даних.

Отже, створено автоматизовану систему управління рухомою платформою, яка надає можливість створювати карту небезпечної зони та використовувати її для автоматизованого і автономного дослідження. Також описано програмну реалізацію та архітектуру системи. Описано її основні модулі, аргументовано вибір технологій. На прикладі показано основні сценарії використання системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Szewczyk, R. Progress in Automation, Robotics and Measuring Techniques. : Volume 2 Robotics / R. Szewczyk, C. Zieliński, M. Kaliczyńska // Springer, 2015. P. 81 – 92
2. Меньков А.В. Теоретические основы автоматизированного управления/ А.В. Меньков, В.А. Острейковский. – Учебник для вузов. – М.: Издательство Оникс, 2005. – 640 с.
3. Харазов В.Г. Интегрированные системы управления технологическими процессами. – СПб.: Профессия, 2009. – 592 с.
4. Береза А. М. Основи створення інформаційних систем: навч. посіб. / А. М. Береза. – 2 вид., перероб. і доп. – К.: КНЕУ, 2001. – 214 с.
5. Основи інформаційних систем: Навч. посібник. – Вид. 2-ге, перероб. і доп. / В. Ф. Ситник, Т. А. Писаревська, Н. В. Єрємін, О. С. Краєва; За ред. В. Ф. Ситника. — К.: КНЕУ, 2001. — 420 с.
6. ДСТУ 2226-93 Автоматизовані системи. Терміни та визначення. – К.: УкрНДІССІ, 1994. – 92 с.
7. ГОСТ 24.104-85 Единая система стандартов автоматизированных систем управления. Автоматизированные системы управления. Общие требования. – М.: ИПК Стандартов, 2002. – 14с
8. OrangePi download resources [електронний ресурс] — Режим доступу: <http://www.orangepi.org/downloadresources>
9. Використання високопродуктивних мікроконтролерів для підвищення економічної ефективності виробництва джему / Р. С. Бессмертный, П. Ю. Катін // Стандартизація. Сертифікація. Якість. - 2019. - № 3. - С. 69-77. - Режим доступу: http://nbuv.gov.ua/UJRN/ssia_2019_3_11

10. Katin, P. (2017). Development of variant of software architecture implementation for low-power general purpose microcontrollers by finite state machines. EUREKA: Physics and Engineering, 3, 49–54. Режим доступу: <https://doi.org/10.21303/2461-4262.2017.00361>
11. Бессмертний, Р. С. Програмно-апаратна інфраструктура наземного робота для дослідження території : дипломний проект ... бакалавра : 6.050103 Автоматика та управління в технічних системах / Бессмертний Роман Сергійович. – Київ, 2019. – 76 с.
12. Batteries — choose the right power source for your robot [електронний ресурс] — Режим доступу: <https://medium.com/husarion-blog/batteries-choose-the-right-power-source-for-your-robot-5417a3ec19ca>
13. Зенкевич, С. Л. Система управління мобільного колесного робота / С. Л. Зенкевич, А. В. Назарова // Вестник Московского Государственного Технического Университета им. Н. Э. Баумана, Сер. Приборостроение, 2006. Вып. 3. С. 31 – 51.
14. Choosing the right wireless communication protocol [електронний ресурс] — Режим доступу: <https://neuronicsworks.com/blog/wireless-communication-protocols/>
15. How to choose components to build a basic robot [електронний ресурс] — Режим доступу: <https://hub.packtpub.com/how-to-choose-components-to-build-a-basic-robot/>
16. L298N H-bridge datasheet [електронний ресурс] — Режим доступу: https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf
17. STM32F103C8T6 Datasheet (PDF) - STMicroelectronics [електронний ресурс] — Режим доступу: <https://pdf1.alldatasheet.com/datasheet-pdf/view/201596/STMICROELECTRONICS/STM32F103C8T6.html>

18. Pros and Cons of Using Spring Boot [электронный ресурс]:
<https://scand.com/company/blog/pros-and-cons-of-using-spring-boot/>
19. A java package to access the Capture interface of the Video4Linux API
[электронный ресурс]: <https://code.google.com/archive/p/v4l4j/>
20. A Look at SQLite (And How It Could Benefit You) [электронный ресурс]:
<https://www.a2hosting.com/blog/sqlite-benefits/>
21. Analysis of How Mobile Robots Fail in the Field – Jennifer Carlson – University
of South Florida, 2004 – 70с
22. Вентцель Е.С., Овчаров Л.А.. «Теория случайных процессов и ее
инженерные приложения». - 2003. – 177 с